STRUCTURED REQUIREMENTS DETERMINATION FOR INFORMATION RESOURCES MANGEMENT

THESIS

Tamara C. Mackenthun, B. S.
Captain, USAF

AFIT/GIR/LSQ/88D-8

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY
Wright-Patterson Air Force Base, Ohio

(2)

AFIT/GIR/LSQ/88D-8

STRUCTURED REQUIREMENTS DETERMINATION FOR INFORMATION RESOURCES MANGEMENT

THESIS

Tamara C. Mackenthun, B. S.
Captain, USAF

AFIT/GIR/LSQ/88D-8

The contents of the document are technically accurate, and no sensitive items, detrimental ideas, or deleterious information is contained therein. Furthermore, the views expressed in the document are those of the author and do not necessarily reflect the views of the School of Systems and Logistics, the Air University, the United States Air Force, or the Department of Defense.

AFIT/GIR/LSQ/88D-8

STRUCTURED REQUIREMENTS DETERMINATION FOR INFORMATION RESOURCES MANAGEMENT

THESIS

Presented to the Faculty of the School of Systems and Logistics
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Information Resources Management

Tamara C. Mackenthun, B.S.
Captain, USAF

December 1988

Preface

The purpose of this research was to develop a design support system to assist in the requirements determination phase of Information Resources Management System development. Dr. Benjamin Ostrofsky's *Design, Planning and Development Methodology* was the theoretical basis for this work.

The bulk of the programming which was central to this research was carried out in the HyperCard® programming environment. The resulting HyperCard® stack has the capability to launch the Design/IDEF™ diagramming tool to build graphic representations of design concepts.

I would like to thank my advisor, Lieutenant Colonel Dick Peschke, who provided me with the incentive and freedom to strive for a creative, worthwhile thesis. His advice, concern, and friendship made what could have been a painful experience into an extremely rewarding endeavor.

I would also like to extend a thank you to the many others who helped me produce this thesis, namely: Professor Dan Reynolds, whose enthusiastic attitude toward learning has been a constant motivation and inspiration; Lieutenant Colonel Skip Valusek, for placing my ideas on the right track; Captain Maurice Riggins, invaluable all-around Macintosh expert; and Captain Kathy Austin, for volunteering to act as a reader.

I owe a special note of thanks to Captain Fredi Daubard, owner and operator of Fredi's Bistro, Laundromat and Counseling Service. Her friendship, support, and home cooking have kept me going during these eighteen difficult months.

And finally, I thank my family for their inexhaustible patience understanding, and love. Thank you, Michael and Christopher, for enduring marriage and motherhood via long distance telephone; and Mom, for making this separation so much easier on us all.

"You ought to have finished," said the King. "When did you begin?"

The Hatter looked at the March Hare, who had followed him into the court, arm-in-arm with the Dormouse. 'Fourteenth of March, I think it was," he said.

"Fifteenth," said the March Hare.

"Sixteenth," said the Dormouse.

"Write that down," the King said to the jury; and the jury eagerly wrote down all three dates on their slates, and then added them up, and reduced the answer to shillings and pence.

— Lewis Carroll (1865)

Table of Contents

Page

List of Figures

Figure	Page

List of Tables

Abstract

The purpose of this ~~research~~ thesis was to provide the Information Resources Management System designer with a framework on which to structure the decisions which must be made in order to translate rapidly changing information needs into plans for Information Resources Management Systems which implement rapidly changing technology.

The HyperCard® programming environment and the Design/IDEF™ diagramming tool were used to develop a design support system which guides the Information Resources Management (IRM) system designer through the requirements determination stage of Dr. Benjamin Ostrofsky's *Design, Planning and Development Methodology.* This system consists of the Design, Planning and Development (D, P &D) Stack, a Help stack, and a User's Manual. The system guides the IRM system designer through the requirements determination process, assists in the collection of data, and organizes that data into a form which can be subjected to objective analysis and optimization.

The system currently supports only the requirements determination phase of a complete Information Resources Management System design methodology. It is intended to serve as input to future development of a complete sytem to assist the Information Resources Management System designer with all phases of the design process. (KR)

STRUCTURED REQUIREMENTS DETERMINATION FOR INFORMATION RESOURCES MANAGEMENT

I. *Introduction*

> The White Rabbit put on his spectacles. "Where shall I begin, please your Majesty?" he asked.
>
> "Begin at the beginning," the King said, very gravely, "and go on till you come to the end: then stop."
>
> — Lewis Carroll (1865)

Background

During recent years, managers have recognized that as much as 80% of their time is spent in the collection, processing and communication of information (Davis and Olson, 1985:4). This recognition has brought home the importance of effective and efficient management of information to the successful operation of the organization. As a result, managers are now actively engaged in seeking methods to improve their information management capacity which surpass the intuitive and time consuming procedures of the recent past. Advances in information processing capability, as a result of new developments in computer technology, are causing managers to turn to the computer for assistance in managing information (Diebold, 1985:4).

Computers were first introduced in organizations to take over the easily routinized, highly repetitive tasks such as accounting and inventory; commonly referred to as transaction processing applications (Tom, 1987:305; Davis & Olson, 1985:132). In the last 25 years, the operational capabilities and features of computers have grown exponentially. Computers are now appreciably smaller and faster, have far

greater storage capacity, and are substantially cheaper (Davis, 1988:2). This decrease in the cost and size of computer hardware, with its associated increase in operational capability, has expanded the scope and increased the number of tasks which can be taken over or augmented by the use of computers. As a result, increasing numbers of managers are using computer systems to support the "fuzzy" decision making problems associated with managerial control and strategic planning. This evolution of computers from machines suitable only for transaction processing to tools with the potential to support activities at all levels of the organization has changed the way businesses view the utility of computer technology. This shift is characterized by a decreasing emphasis on the sheer amount of data processed or the number of reports printed and distributed to a focus on the quality and value of computer output for decision making (Synott & Gruber, 1981:3). This shift has prompted designers of computer systems to move from a mindset focused on hardware and transaction processing of data to the decision-support based, information orientation of the future (Synott & Gruber, 1981:4).

This change from an orientation on unprocessed facts to an emphasis on information useful for decision making has forced managers to discriminate between the concepts of "data" and "information". Data are simply undeveloped, raw facts (Peschke, 1985:2). Information is the result of processing data in order to gain insights from which to make decisions and take actions (Bryce, 1983: 88). This "processing" is essentially the formation of conceptual links, or connections, between data items (Hoffman, 1980:293). A more concise way of representing this concept is with Hoffman's (1980:293) formula:

Information = Facts, Figures, + their meaningful connections

Thus, an **Information System** must involve more than simply reporting data, it must also help the user process that data and help form these meaningful connections in order to produce information. This

distinction requires an expansion of the definition of Information System beyond the hardware oriented definitions associated with the more basic transaction processing systems. A definition which recognizes the more expansive way organizations are viewing, managing and using information is:

> An information system is a combination of people, equipment, facilities, procedures, and other resources that are organized for the purpose of, but not limited to, creating, collecting, protecting, analyzing, storing, retrieving, and disposing of information [Peschke, 1985:3].

This definition describes information in the same terms used to define the traditional assets, or resources, of an organization: manpower, material, money and machines. Effective management of information requires that organizations recognize that information is also a resource, and that it must be treated in the same manner as the traditionally recognized resources — it must be managed and controlled (Diebold, 1985:41, Peschke, 1985:4).

Treating information as a resource is not a simple task. A manager cannot simply apply the time honored techniques for managing and controlling the traditional resources to the management and control of information Information has many qualities which distinguish it from the traditional resources. Information is abstract; one cannot hold it or even gain a mental picture of its makeup. Information is non-exhaustive, it does not deplete. Information is self-generating; it expands as it is used and new conceptual links are formed, and the more a chunk of information is used, the more it expands (Naisbitt, 1982). Information is quickly transported or copied, which allows it to be easily shared (Cleveland, 1982:37). These characteristics make it impossible to apply traditional economic theories to the information resource.

These unique properties, combined with the necessity for treating information as a resource, have led to the creation of a new field of management devoted to finding ways to manage information as a resource:

Information Resources Management (IRM). Dr. Elizabeth Byrne Adams, a Professor of Management at George Washington University, defines this new field as:

> ... a management function to develop and implement policies, programs, and guidelines to plan for, manage and control ... information resources (Adams, 1980).

An important aspect of Professor Adams' definition is that it does not mention computers or automated systems. This omission emphasizes the fact that computers are just one of the many tools which are used in the management of the information resource. IRM encompasses the management of information itself and other resources, such as personnel, equipment, funds and technology used in the process (FIRMR, 1985). A system which assists in the management of these elements, an **Information Resources Management System**, can be defined as "the set of activities which is concerned with the systematic management of all the resources used *in the process of managing information*" (Peschke, 1985:4).

IRM Systems are intended to unify processes and procedures for dealing with the acquisition, standardization, classification, inventory, dissemination, and use of information of every kind, throughout the organization (Diebold, 1985:47). Because the primary emphasis of IRM is on treating information as a major organizational resource, the primary level of implementation of IRM systems is at the level of the Organizational Information System (OIS) (Peschke, 1985:10). The OIS encompasses all tools which are used in an organization to manage information in all its forms (Figure 1). It is the focal point where all information processing activities in the organization come together (Siegel, 1975).

As we proceed into the 1990s, the primary objective of IRM professionals will be to find methodologies which they can use to construct IRM systems for managers at the OIS level who consider information a vital

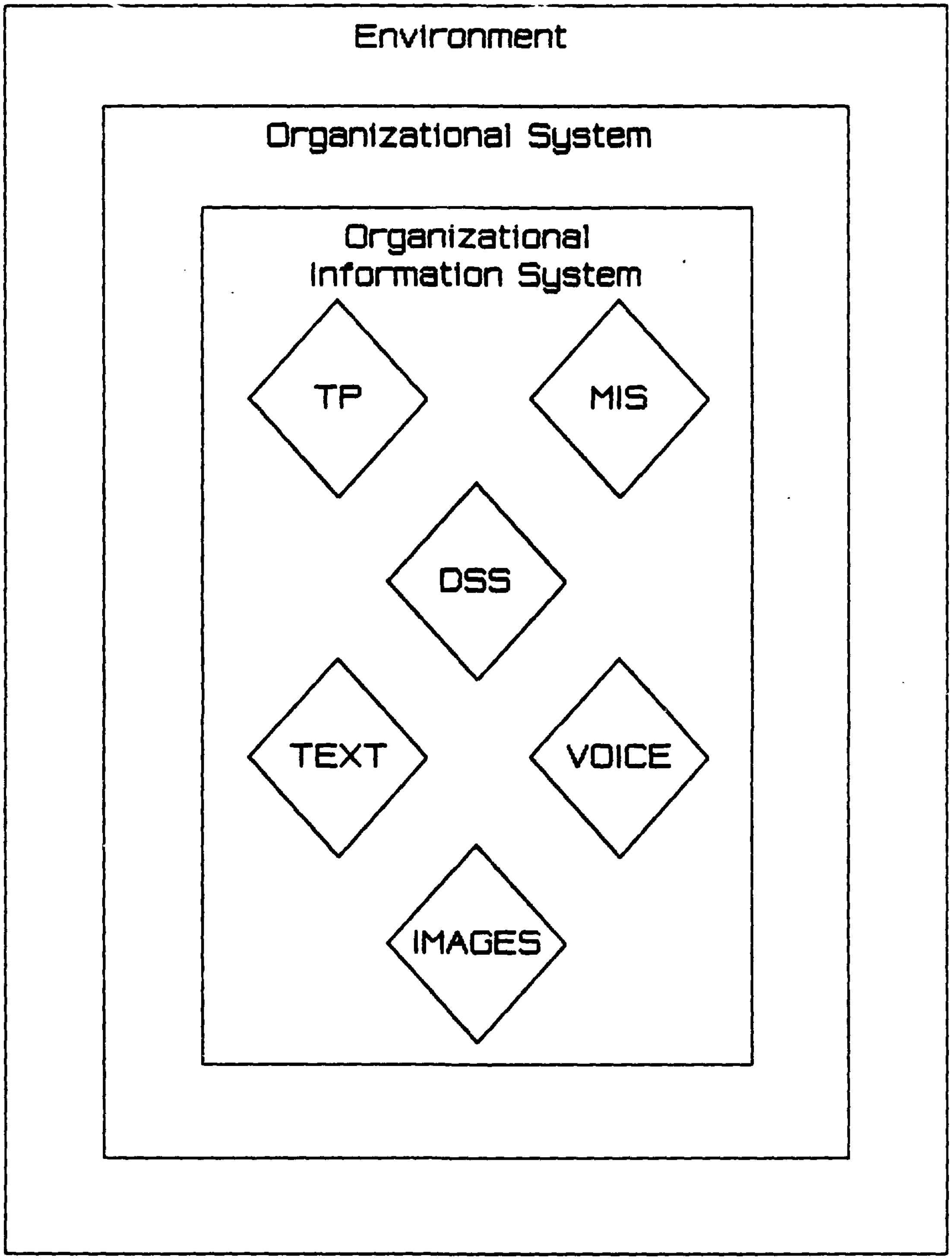


Figure 1. The Organizational Information System
(Siegel, 1975:2; Peschke, 1985:11)

organizational resource (Bryce, 1983:89). However, because of the unique properties of information, methodologies for IRM system design are inherently more complex and difficult to structure than the already complicated methodologies used for designing systems for the management of traditional resources.

Structuring the IRM Design Problem

Current methods for designing IRM systems do not provide the structure and discipline necessary to ensure the resultant system will effectively manage and control an organization's information (Davis, 1982; Yadav, 1983; Peschke, 1985). Organizations are plagued with complaints about information systems that do not meet the needs of users, are not easily adaptable to ever-changing requirements, and which cost more and take longer to implement than anticipated (Bryce, 1983:88).

Milton Bryce, one of the first computer programmers in the US, blames these problems on the lack of a good methodology with the organization, structure and discipline necessary to consistently design and build good IRM systems. He hypothesizes that without a sound, standard IRM system design methodology, managing information as a resource will be "a corporate pipe dream" (Bryce, 1983:88).

Current State of the Art

Existing information system design methodologies do not provide this complete structure. Most information system design procedures separate the issue of determining the information requirements of an organization from the establishment of the design requirements (Yadav, 1983). Additionally, many methods for designing information systems provide structured techniques for determining needs (Booth, 1983:4; Colter, 1984:51); however, they do not define a structure for organizing this data, for making the decisions about which information items are more

important than others, or for deciding which system of the ones considered is the relative best (Bryce, 1987:89).

Peschke considered these criteria in his review of seven of the predominant methods available for information requirements determination. His results are shown in Table 1. Peschke concluded that there is no specifically developed, sound structured system design methodology which starts with assisting the IRM system designer in determining requirements and continues to structure the problem up through the establishment of design specifications (Peschke, 1985:40).

Further research needs to be conducted in the area of requirements determination and system design methodology in order to alleviate this piecemeal approach to designing IRM systems.

Specific Problem

The problem addressed by this research is the development of an IRM system design structure. This structure will provide the discipline to support the design and implementation of information systems which consistently meet the needs of a using organization.

Research Objective

The objective of this research is to develop a computer based design tool which will provide the designer with a structured framework to accomplish the requirements determination phase of IRM system design within the context of a complete IRM system design methodology. The requirements determination phase is the most complex phase of the design process (Ostrofsky, 1988). There is a large amount of information which must be collected, synthesized and acted upon. The system developed in this research will assist the designer by functioning as an electronic notebook which documents those items which must be considered when making design decisions, with the added advantage that the system will assist in the organization and processing of those items. When complete,

Table 1. Summary of Information Requirements Determination Procedures (Peschke, 1985:38)

	Organization-Level Analysis	Documented Output	Supports Conceptual Modeling	Addresses Organization Setup	Supports Process Automation	Well Defined Problem Analysis	Defines Information Architecture	Results Obtained Quickly	Address Life-Cycle of the Information Resource	Objective Optimization Procedure
Business Information Analysis & Integration Technique (BIAIT) (Burnstine, 1979)	*	Y	Y	N	Y	Y	Y	Y	N	N
Business Information Control Study (BICS) (Kerner, 1979)	Y	Y	N	N	N	Y	N	Y	N	N
Business System Planning (BSP) (IBM, 1981)	Y	Y	Y	N	N	N	N	N	N	N
Critical Success Factors (CSF) (Benjamin, 1982)	P	P	N	N	N	N	N	Y	N	N
Information Systems Work & Analysis of Changes (ISAC) (Lundberg et al., 1981)	Y	Y	Y	N	N	N	N	N	N	N
Method for Business Analysis & StructuredAnalysis & Design(MBI/SAK) (Wigander et al., 1984)	Y	Y	N	N	Y	Y	Y	N	N	P
Organization Analysis & Requirements Specification Methodology (ORASM) (Yadav, 1981)	P	Y	Y	P	Y	Y	Y	N	N	N

Y = Yes N = No P = Partially * = Must be tailored

this system will guide the designer through the methodology, assist in the collection of data, and will organize that data into a form which can be objectively analyzed and optimized.

This research is intended to serve as input to future development of a full scale, complete, automated methodology which can help the designer accomplish all phases of the IRM system design process.

Assumptions

1. Potential users are familiar with the design methodology on which the research is based.

2. Potential users are familiar with the computer hardware and software used to implement the system.

Additional assumptions which were made during the course of this research effort are described in Chapter Three.

Limitation

The system currently supports only the requirements determination phase of the complete IRM system design methodology.

II. *The Tools for Design*

"A slow sort of country!" said the Queen. "Now here, you see, it takes all the running you can do, to keep in the same place. If you want to get somewhere else, you must run at least twice as fast as that!"

— Lewis Carroll (1872).

Rapid technological progress often outstrips the rate at which technology can be applied to the enhancement of managerial activities (Asimow, 1962:2, Martin, 1987). This problem is especially apparent in the field of Information Resources Management (IRM). As new technological options for enhanced information management are introduced, the possible combinations of hardware, software, and procedures increase exponentially, and the number of choices which the IRM system designer must make increases at the same rate (Simons and Ostrofsky, 1988:49). Information requirements also change rapidly (Davis & Olson, 1985:4; Land, 1982:220). Information that was sufficient to help a manager make a decision yesterday may not be sufficient tomorrow (Andrews, 1983:16).

Because the requirements and technology necessary to build IRM systems are both mercurial, IRM systems designers must maintain a close link between new technology and its application to human needs, and they must be able to conceive of bolder, faster improvements (Valusek, 1988a).

In order to make these improvements, the IRM system designer requires a framework on which he or she can structure the decisions which must be made in order to translate rapidly changing needs into plans which implement rapidly changing technology (Peschke, 1985; Valusek, 1988a). The designer can turn to the fields of engineering design and adaptive design to help formulate the structure required to sequence the decisions which must be made in the process of developing an accurate set of requirements for an IRM system.

Engineering Design

Engineering design strategies are used to structure design decisions when the appropriate technology is complex and its application not obvious, and when the prediction and optimization of the outcome requires analytical procedures (Asimow, 1962:2). Engineering design procedures are intended for explicit decision making in an unstructured problem area in which only "fuzzy" definitions and requirements can be identified (Ostrofsky & Kiessling, 1984:7).

Engineering design strategies typically include three phases: a Feasibility Study, the purpose of which is to develop a set of alternative systems for further study; a Preliminary Design phase, during which the designer determines which of the alternatives is the relative best; and the Detailed Design phase, which includes the developmental activities associated with the implementation of the best system (Asimow, 1962:12-13; Ostrofsky, 1977:17-20). The *Design, Planning and Development Methodology* developed by Ostrofsky (1977) is an implementation of the engineering design philosophy which is particularly well suited to the IRM system design problem (Peschke, 1985:63).

The Design, Planning and Development Methodology

Ostrofsky's *Design, Planning and Development Methodology* is a sequentially structured engineering design strategy capable of evaluating multiple criteria simultaneously and providing the iterative capabilities needed to incorporate additional information (Ostrofsky, 1977). It is a rigorous methodology for identifying, considering, and integrating relevant criteria involved in the design of a complex system (Simons & Ostrofsky, 1988:49). The design-planning methodology recognizes that the number and complexity of design alternatives often make the design problem impossible for the decision maker to resolve without a structured method for organization and evaluation. It also recognizes that an attempt to achieve one objective may conflict with the attempt to achieve another, and

provides concrete analytical tools which help the designer resolve these conflicts.

Ostrofsky structures the design process by first defining the two major phases in the life of any activity; the Production-Consumption Phase and the Primary Design-Planning Phase. These two phases, their constituent activities, and relationship are illustrated in Figure 2.

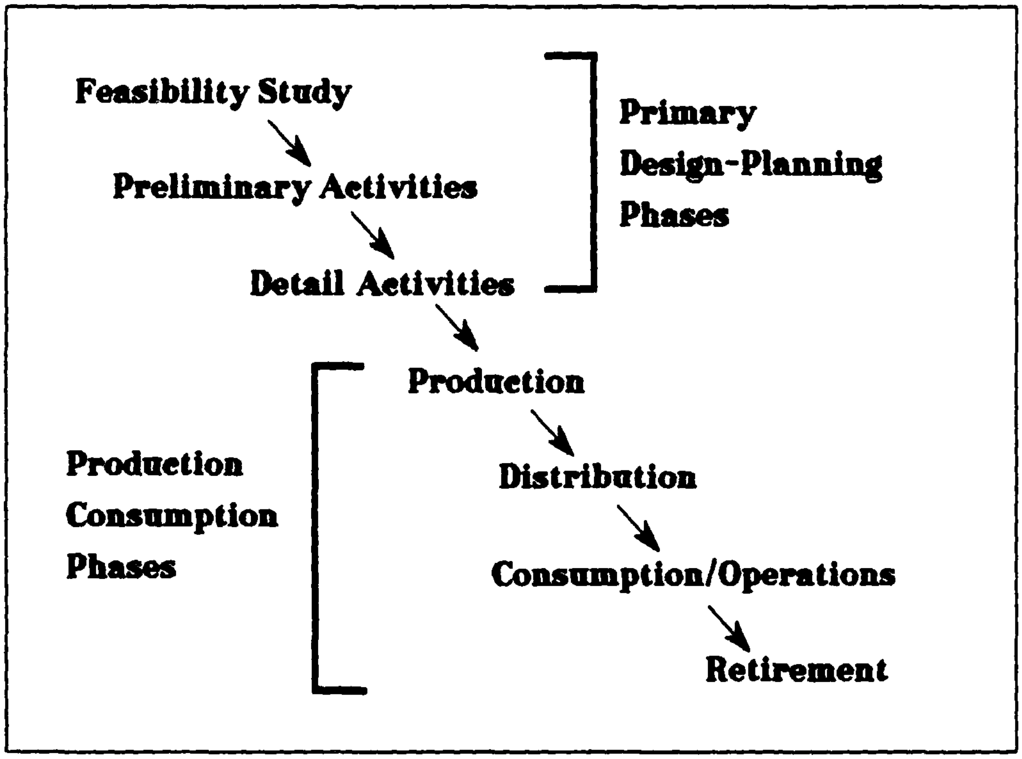


Figure 2. Phases in the Life of an Activity (Ostrofsky, 1977:18)

The Production-Consumption Phase defines the operational life of the activities resulting from the decision maker's actions (Peschke, 1985:53). It is the actual operation of the system within the context for which it was designed. The Production-Consumption Phase consists of the chronological sequence of:

• Production — the activities which produce the system elements, or product.

• Distribution — the activities which flow the raw materials into the production facility, and the product out to the consumer location.

• Consumption/Operation — The use of the elements by the consumer; or, if the product is an operation, the monitoring of that operation.

• Retirement — activities necessary to convert the system to a permanently inactive status (Ostrofsky, 1977:8-9).

The Primary Design-Planning Phase is the time during which the designer works to identify, select and develop plans for a feasible solution which will meet the needs of the user during the Production-Consumption Phase. An explicit requirement of the methodology is that the designer consider the production-consumption phase throughout the design planning phase. This requirement is vital to the successful design of a system; it can be difficult or impossible for the system to reflect a necessary characteristic if that characteristic has not been considered during each of the various decisions which must be made from the beginning of the design process (Simons & Ostrofsky, 1988:49). The three major elements of the Primary Design-Planning phase are:

• Feasibility Study, which results in the generation of a set of useful, or candidate systems.

• Preliminary Activities, which identify the optimal, or relative best, system from the set of candidate systems identified in the Feasibility Study.

• Detail Activities, which are the developmental activities associated with formulating the concrete plans for the implementation of the optimal system (Ostrofsky, 1977:155).

The activities in the feasibility study are the focus of this research. The feasibility study consists of four sequential phases (Figure 3).These four phases are:

• Needs Analysis. The product of the needs analysis is a general statement of the project. This statement both defines the direction of

subsequent activities and justifies the further expenditure of resources on its continued development (Simon & Ostrofsky, 1988:50).

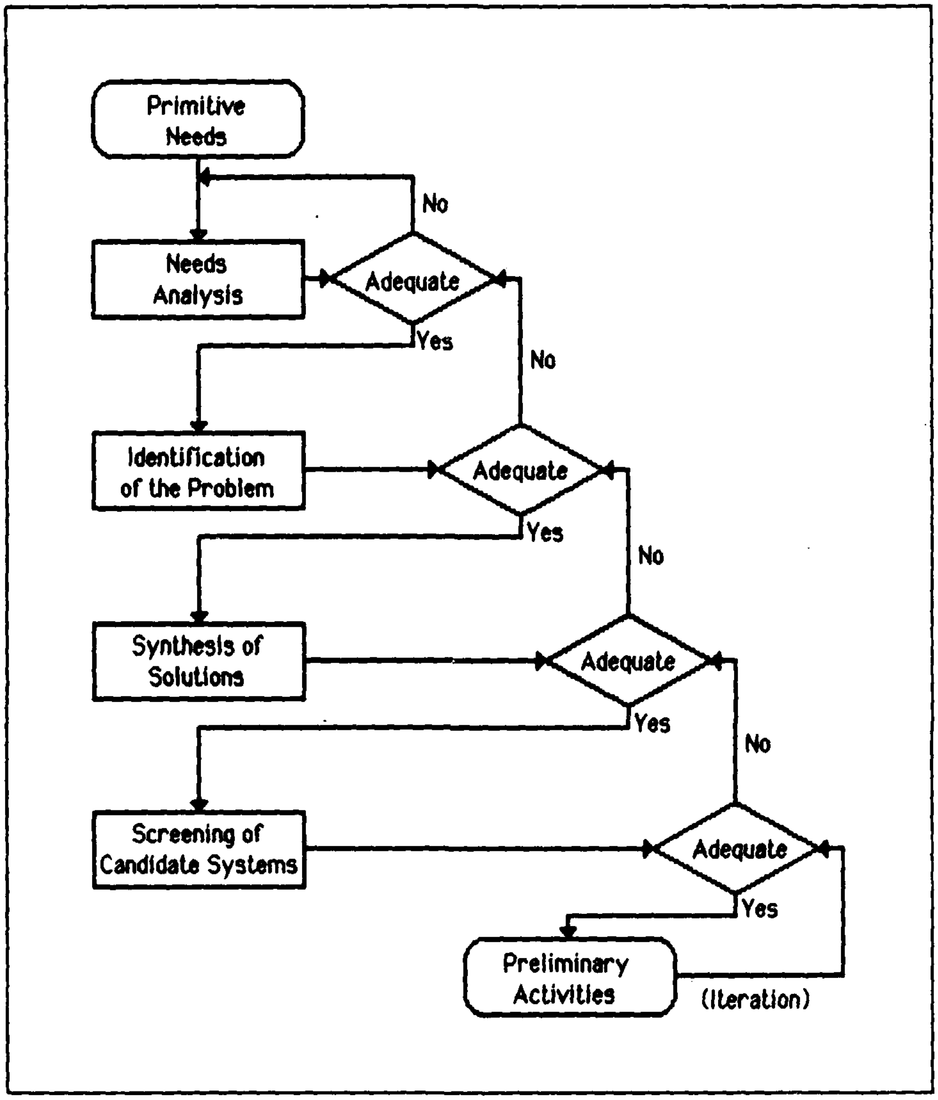


Figure 3. Feasibility Study Activities (Ostrofsky, 1977:28)

• Identification of the Problem. During this phase, the needs are placed in the context of the production-consumption phase, and are analyzed in terms of inputs to and outputs of the system. This is accomplished using a

matrix as shown in Figure 4. It is during this phase that the designer makes an explicit connection between the activities in the primary design planning phase and those in the production-consumption phase.

	INPUTS		OUTPUTS	
	Intended	Enviornmental	Desired	Undesired
Production				
Distribution				
Consumption/ Operation				
Retirement				

Figure 4. Input/Output Matrix (Ostrofsky, 1977:36)

• Synthesis of Solutions. The synthesis of solutions consists of structuring concepts, basic approaches to the problem, and breaking these concepts down into their elemental functions, or subsystems. The designer then formulates as many alternative ways to accomplish each function as possible, and candidate systems are created by combining one alternative for each subsystem. The relationships between concepts, subsystems and candidate systems are shown in Figure 5.

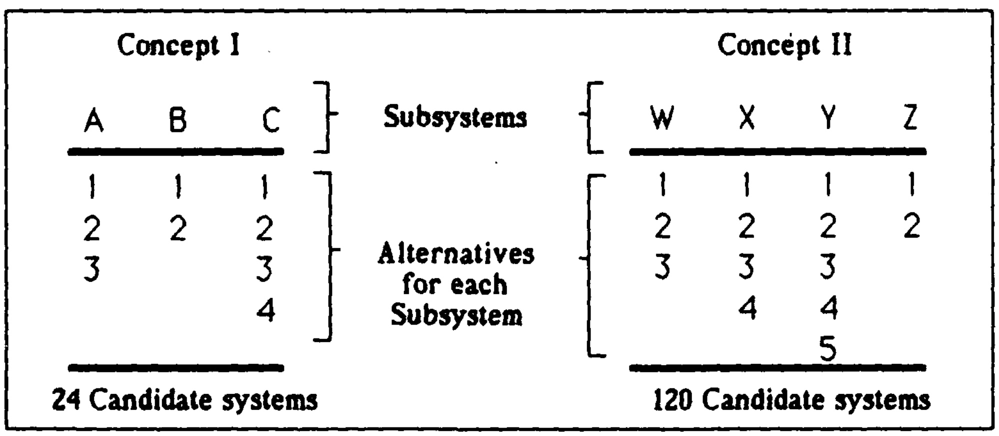


Figure 5. Concept, Subsystem, Alternative and Candidate System Relationship (Ostrofsky, 1977:48)

• Screening of Candidate Systems. The candidate systems are screened to eliminate those which are clearly physically, economically, or financially infeasible.

The Design Planning Methodology Applied to IRM System Design

Lt Col Richard Peschke, currently a professor at the Air Force Institute of Technology, developed a structured optimization method for IRM system design using Ostrofsky's *Design, Planning and Development Methodology* (Peschke, 1985). Peschke focused on the elements of the preliminary activities phase as they applied to IRM system design, and demonstrated the application of the methodology to information requirements analysis using data similar to what the designer-planner would actually have available (assuming the feasibility study had been accomplished) (Peschke, 1985:15). He demonstrated that the design planning methodology was well suited to the IRM system design problem, and that candidate IRM systems can be successfully subjected to rigorous analysis to determine which candidate system would best meet the needs of the organization (Peschke, 1985:138).

Peschke's methodology ensures the IRM system chosen from a group of candidates will "best" meet the needs of the using organization. A complete methodology would include a strategy for formulating the candidate systems which could then be subjected to this analysis. This requires the adaptation of the feasibility study, or elicitation of needs, phase of Ostrofsky's methodology to the IRM system design problem.

Structuring Information Requirements Determination

The primary purpose of the requirements determination process for information systems is to identify the information which a worker needs in order to perform his or her job (Lederer, 1981:15). The success of an IRM system may well depend on the correctness of the results of the requirements determination process, yet traditionally no more than 25% of

the resources needed for system development have been spent on defining and analyzing candidate systems (Seagle & Belardo, 1986:12).

Ostrofsky's methodology provides an excellent framework for organizing and classifying information requirements for analysis, but does not provide explict methods for eliciting information requirements from the eventual users of the system. This is because much more is known about how to model a decision and provide decision makers with appropriate information once the conditions and their relationships have been identified than about the process of collecting the relevant data (Montazemi & Conrath, 1986:46). Because of this lack of a structured framework for requirements elicitation, the eventual users and the designers of IRM systems are not able to effectively and consistently deal with the complex and conflicting objectives involved in the determination of requirements for IRM systems (Bowman, 1963).

Requirements can be viewed as a person's representation of needs, where a need is a gap between existing and desired conditions. The extent to which a requirement for an IRM system accurately represents a need determines the quality of the resulting system (Kauffman & English, 1979; Valusek & Fryback, 1985:104). There are two obstacles in the construction of a representation within the context of the information requirements determination problem: the constraints on humans as information processors and problem solvers, and the complex interactions between users and designers in defining requirements (Davis, 1982; Valusek & Fryback, 1985).

Humans as Information Processors. The eventual users of IRM systems often do not know what they want the system to do for them, mainly because they are unable to accurately describe how they do their job (Ackoff, 1967:B-149; Montazemi & Conrath, 1986:45; Robey & Taggart, 1983:278). People are biased toward identifying requirements which are based on current procedures, currently available information, recent events, and inferences from small samples of events (Valusek & Fryback, 1985:104; Davis, 1982:7; Simon et al, 1987:19). Interviews, the assumed method for

obtaining information about the process to be supported when using Ostrofsky's methodology for information system requirements determination, are not equipped to deal with these human information processing deficiencies (Valusek & Fryback, 1985:105).

Interaction Between Users and Designers. The already difficult task of eliciting needs from users is compounded by the fact that designers of information systems are often not familiar with the process which the system will support. Because of this unfamiliarity with the underlying process, the designer often does not know what questions to ask (Bryce, 1983:90). Additionally, the designer and the user will often have different perceptual frameworks which will tend to bias their perceptions when attempting to communicate (Valusek & Fryback, 1985:106). Because of these interaction problems, the designer needs a methodology which will help discover what information users need to do their job without having to directly ask the question (Lederer, 1981:15).

An area of study which could help add structure to the elicitation process for IRM system needs within the framework of Ostrofsky's methodology is the field of Adaptive Design for Decision Support Systems.

Adaptive Design

Adaptive Design refers to procedures which allow for design and development of Decision Support Systems (DSS) at the user's location, and at their convenience (Valusek, 1988:106a). DSS designers are faced with the difficult task of determining who to select as a representative user and the necessity for taking this person, who is often vital to the organization, away from his or her work during the time needed to elicit requirements (Valusek, 1988a). Adaptive design techniques are based on the premise that the elicitation of needs should be completed at the user's convenience, not the builder's. Additionally, the techniques allow all users to participate, the designer does not have to search for a representative user (Valusek, 1988b:106). Adaptive design techniques provide a structure which allows

the users to work on the problem of defining requirements in odd moments, when an idea strikes them; the information requirements determination process does not have to be scheduled time away from the workplace or with an interviewer (Valusek, 1988b:107).

Adaptive design techniques help place responsibility on the users by giving them the tools to shape their own system (Valusek, 1988b:106). This forces the user to remain accountable for the eventual success or failure of the system, and can help ensure its success (Bryce, 1983:90).

Adaptive Design Contributions

The actual process of defining needs, analyzing activities by filling in the Input/Output Matrix and generating concepts is an essentially creative process, and as such, it is difficult to assist and structure (Ostrofsky, 1988). There are some specific techniques which can be borrowed from the adaptive design process to enhance this creativity.

Input/Output Matrix. The Input/Output matrix is the most structured aspect of Ostrofsky's feasibility study; however, it is difficult to relate the definitions of the production-consumption phase to the IRM system design problem. This makes it difficult to use as a method for obtaining information concerning needs directly from users. When the user is able to identify a need, if he or she is unable to categorize it immediately within the context of the Input/Output matrix that idea may be lost.

One possible way to assist with this process is to adapt the concept of the "hook book" (Valusek, 1988b:109) to the activity analysis problem. The hook book, as described by Valusek, consists of automated note cards which help users retain thoughts about how to improve an existing system. A hook book entry contains four pieces of information (Figure 6).

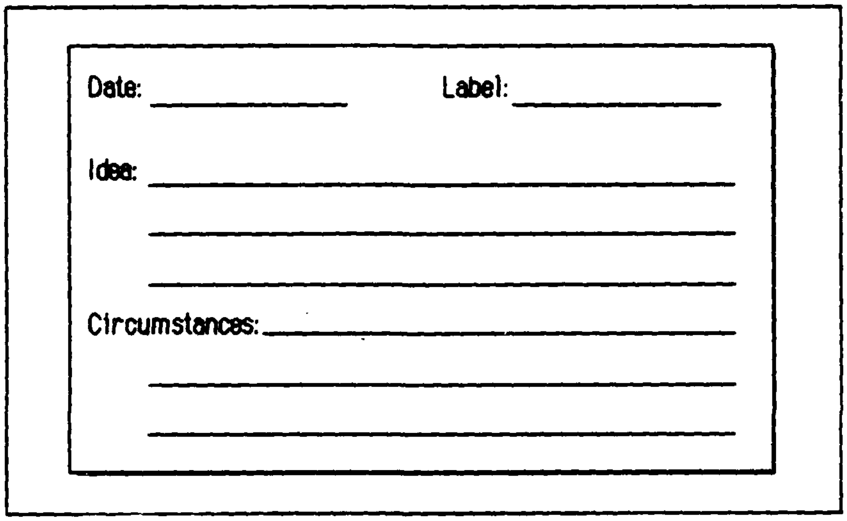


Figure 6. Hook Book Entry Format (Valusek, 1988:109)

These are: the Date, which allows for chronological sorting; the Label, added later, which allows for task sorting; the Idea itself, which is a cursory note about the idea; and the Circumstances, which provides a trigger to detailed recall of the idea during requirements elicitation (Valusek, 1988b:109).

The hook book can also be used for structuring the process of determining initial information system requirements. It can be used to capture the initial thoughts a user has about what he or she would like for an information system to be able to do, and would allow the user to categorize and feed this information into the Input/Output matrix at a later date. It can also be used as a means for users to communicate the relative importance of a need, providing the designer with more information with which to determine the relative weights during the analysis of the candidate systems.

Concept Structuring. For structuring concepts, Ostrofsky's methodology relies on schematic means, such as signal flow charts or functional flow diagrams, for representing system activities (Ostrofsky, 1977:46). Flow diagrams which describe the processes to be supported by an IRM system are complex and difficult to design, primarily

because the designer is faced with an unstructured decision environment in which to translate stated needs into activities of a future system (Montazemi & Conrath, 1986:45). A technique which would help designers develop a subjective representation of the relationships between factors in such an unstructured environment must be able to capture complexity, peculiarity, idiosyncrasy, generality, and particularity; and represent these aspects in a comprehensible manner (Eden, 1979:53).

A technique which has been shown to meet these requirements, and which could be useful in the context of the *Design, Planning and Development Methodology* is Concept Mapping (Montazemi & Conrath, 1986:45; McFarren, 1987, Valusek, 1988b:107). A concept map is a representation of the relationships which are perceived to exist among the elements of a given environment (Montazemi & Conrath, 1986:46). Concept maps are used to externalize concepts and propositions, and to negotiate meanings between users and designers (Novak & Gowin, 1984:20). An example concept map is shown in Figure 7.

The guidelines for concept mapping are very simple. Concept maps are intended to represent meaningful relationships between concepts in the form of propositions. Propositions are two or more concept labels linked by works in a semantic unit. A concept map in its simplest form could be: sky — is — blue (Novak & Gowin, 1984:15).

Concept maps are remarkably effective tools for showing misconceptions (Novak & Gowin, 1984:20). Because of this property, a designer can make a concept map, show it to a user, and the user should be able to identify the designer's misconceptions about the process. Concept maps can also be used in the development of concept representations by helping users or designers obtain an initial conceptualization of the process to be supported by the resultant IRM system.

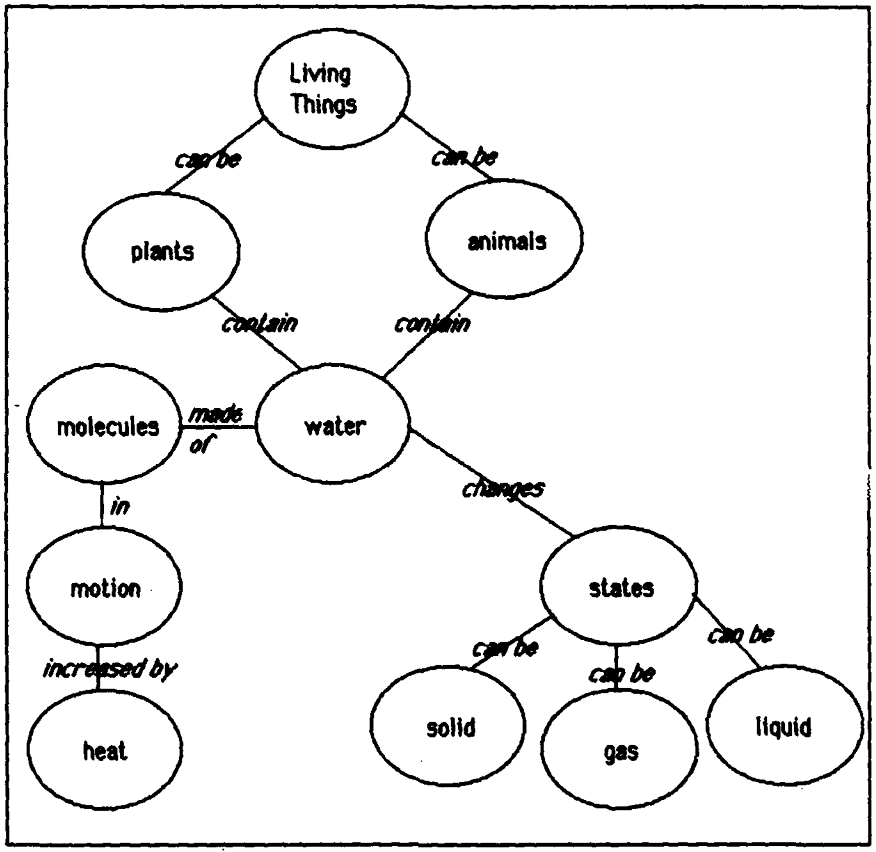


Figure 7. Illustrative Concept Map (Novak & Gowin, 1984:18)

The Development Medium

The system which will result from the adaptation, synthesis, and automation of these diverse tools for information requirements determination will be a system which will help managers make decisions about the desired qualities of IRM systems. Keen and Wagner (1979) propose that a medium which would be suitable for development of such a complex decision support system should provide:

- A flexible development language that allows rapid creation and modification of systems for specific applications.
- A system design architecture that allows quick and easy extensions and alterations.

• An interface that buffers users from the "computer" and allows a dialogue based on the manager's concepts, vocabulary, and definition of the decision problem.

• Communicative display devices and output generators.

One development medium which comes close to fitting this definition is HyperCard®, Apple Computer's development tool for information management systems. HyperCard® is an implementation of the concept of Hypertext.

Hypertext. Hypertext is a method for organizing information which provides capabilities for accessing data which are very different from those used by traditional database management systems (Peschke & Austin, 1988:69).

Hypertext had its origins in Vanever Bush's studies concerning the mechanization of associative memory structures. Bush observed that in most reference systems data are stored alphabetically or numerically, and in order to find a specific piece of data the user must trace down hierarchically from subclass to subclass. The problem with this organization is that the human mind does not organize information hierarchically, but by association (Bush, 1945). Bush theorized that if selection by association could be mechanized, people would have a machine which would closely model, and act as an enlarged, intimate supplement to, human memory (Bush, 1945). Advancements in computer technology have begun to form Bush's vision into reality in the form of hypertext.

Hypertext is a computer-supported medium for information in which pieces of information in one or more documents are tied together with specific links, rather than with the hierarchical structures common in conventional documents or databases (Smith & Weiss, 1988). Links can be created according to any criteria, and do not have to comply with the rules common to conventional databases. The links may be directly activated by a pointing device such as a mouse, which causes the document referenced

by the link to appear instantly in a new window on the screen (Conklin, 1987).

The various developmental tools which implement the hypertext concept are providing new ways to integrate and access information resources (Smith and Weiss, 1988:818).

HyperCard®. HyperCard® is Apple Computer's implementation of the hypertext concept. Bill Atkinson, the creator of HyperCard®, describes it as "...an authoring tool and an information organizer." HyperCard® is a medium which facilitates the sharing of information by making it easier for the non-programming specialist to build his or her own complex structures for information management (Goodman, 1987:12; Kahler, 1988:xiii-xix).

HyperCard® is an object-oriented programming environment. In traditional, procedurally-oriented programming environments, data is separate from the objects that operate on it. The program begins at the first statement containing a verb and proceeds (Shafer, 1988:12; Vaughn, 1988:267). Object-oriented programming is based on objects, which are single programming entities consisting of both data and the procedures or functions that operate on that data (Shafer, 1988:12). In an object-oriented programming environment, programs begin when the user activates an object.

There are five classes of objects in HyperCard®: buttons, fields, cards, backgrounds, and stacks (Figure 8).

The HyperCard® environment makes generous use of visual and conceptual metaphors to help users understand these objects and how they are used (Vaughn, 1988:67). HyperCard® objects are presented metaphorically as commonplace objects which mimic objects encountered in daily life. HyperCard®'s dominant metaphor is the index, or file card, and the stacks into which they are organized.

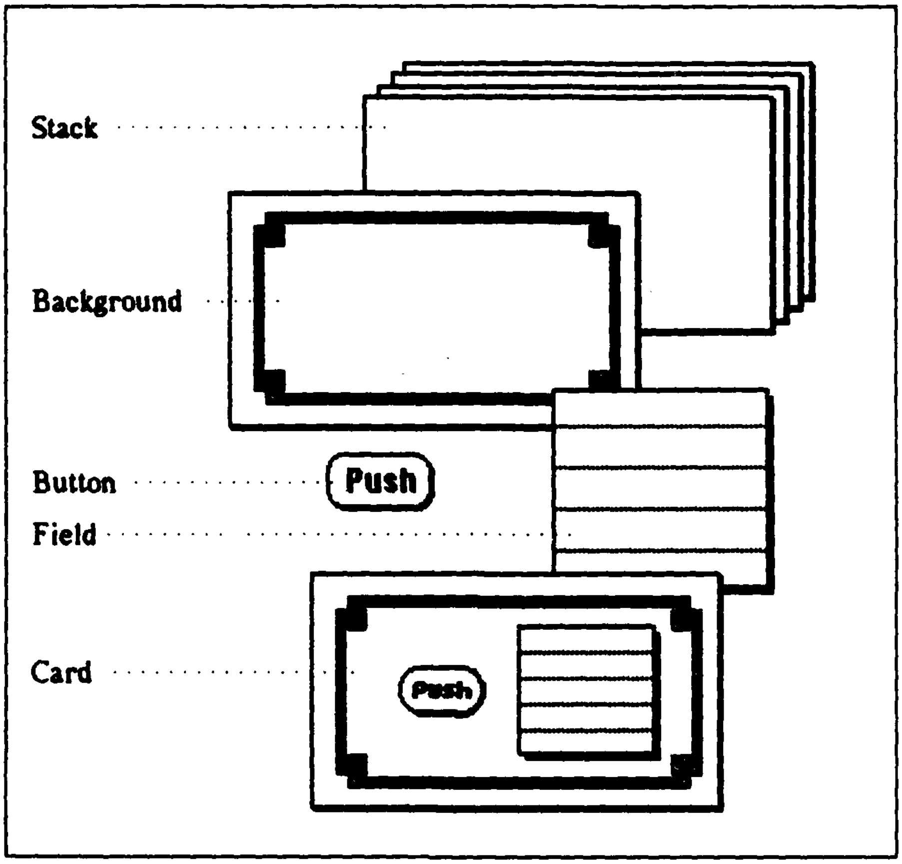


Figure 8. HyperCard® Objects (Apple Computer, 1988:3)

The metaphorical card is the basic element of information in HyperCard®. When you look at the screen of a computer running the HyperCard® program, you see a card. Each card is associated with one background, and a background can be shared by many cards. The information which is specific to a single card overlays the information which is common to all cards of that background, just as the words written on an index card overlay the lines printed on it.

Cards are grouped into stacks, analogous to a stack of index cards. Each stack is stored as a separate file by the computer.

Information can be placed on cards in fields. Information in fields is editable text, and can be processed just as information is processed by a database management system.

Cards can also contain buttons. Buttons are screen areas which can be metaphorically "pushed", or activated, using the computer's mouse. When a button is activated, it sends a message which causes an action, such as searching for a specific piece of information or sorting the cards in a stack.

These messages are specified by programs written in the HyperTalk™ programming language, a relatively simple programming language which is surprisingly similar to English. Programs in the HyperTalk™ language are composed of small chunks, or scripts, each of which is associated with a HyperCard® object. Because a HyperTalk™ script is associated with a specific object, and not a part of a long string of computer procedures, scripts are self-contained, separable elements. This allows a script to be free-standing, and thus easier for the non-programmer to understand and manipulate.

Conclusions

Ostrofsky's *Design, Planning and Development Methodology,* combined with an adaptation of a hook book and using concept mapping to achieve an initial conceptualization of the concept representations could be a powerful methodology to use for determining IRM system requirements. The automation of such a methodology using the HyperCard® development medium will allow the designer to focus on the decisions which must be made, not on the tools being used. The remainder of this thesis describes the development of a system to meet these requirements.

III. *System Design Concept*

"No wise fish would go anywhere without a porpoise"
— Lewis Carroll (1865)

The Stack Design Plan

HyperCard® is a new software environment which is difficult to define in terms of conventional software categories. HyperCard® is described as, among other things, a database management system, an operating system shell, a programming language, and an information organizer (Goodman, 1987; Vaughn, 1988; Shafer 1988; Kaehler 1988). As a result of HyperCard®'s novelty, little guidance exists which directly addresses the planning and development of a HyperCard® stack.

Vaughn (1988:271) and Shafer (1988:6) describe similar strategies for HyperCard® stack design and construction which recommend the following activities be accomplished:

• Conceptualize—analyze the problem, define the data involved, describe the output desired, and break the problem into it's component parts.

• Structure—sketch the background(s), define how users will navigate from one card to another, and build the navigational mechanisms.

• Implement—write the HyperTalk™ computer code to carry out the procedures.

• Finalize —fill in the graphic details.

As a stack is built, the concept and structure change, and this drives changes in the scripts and the graphics.

Design Activities

During the course of building a HyperCard® stack to support the Feasibility Study activities outlined in the *Design, Planning and*

Development Methodology (Ostrofsky, 1977:28), the design activities outlined by Vaughn and Shafer were carried out in the following manner:

Conceptualizing the stack. The initial conceptualization of the stack was formed using the steps outlined in the *Design, Planning and Development Methodology* itself. An Input/Output Matrix was completed in order to describe the needs and bound the problem. The information contained in this matrix was mapped into an initial stack concept (Figure 9).

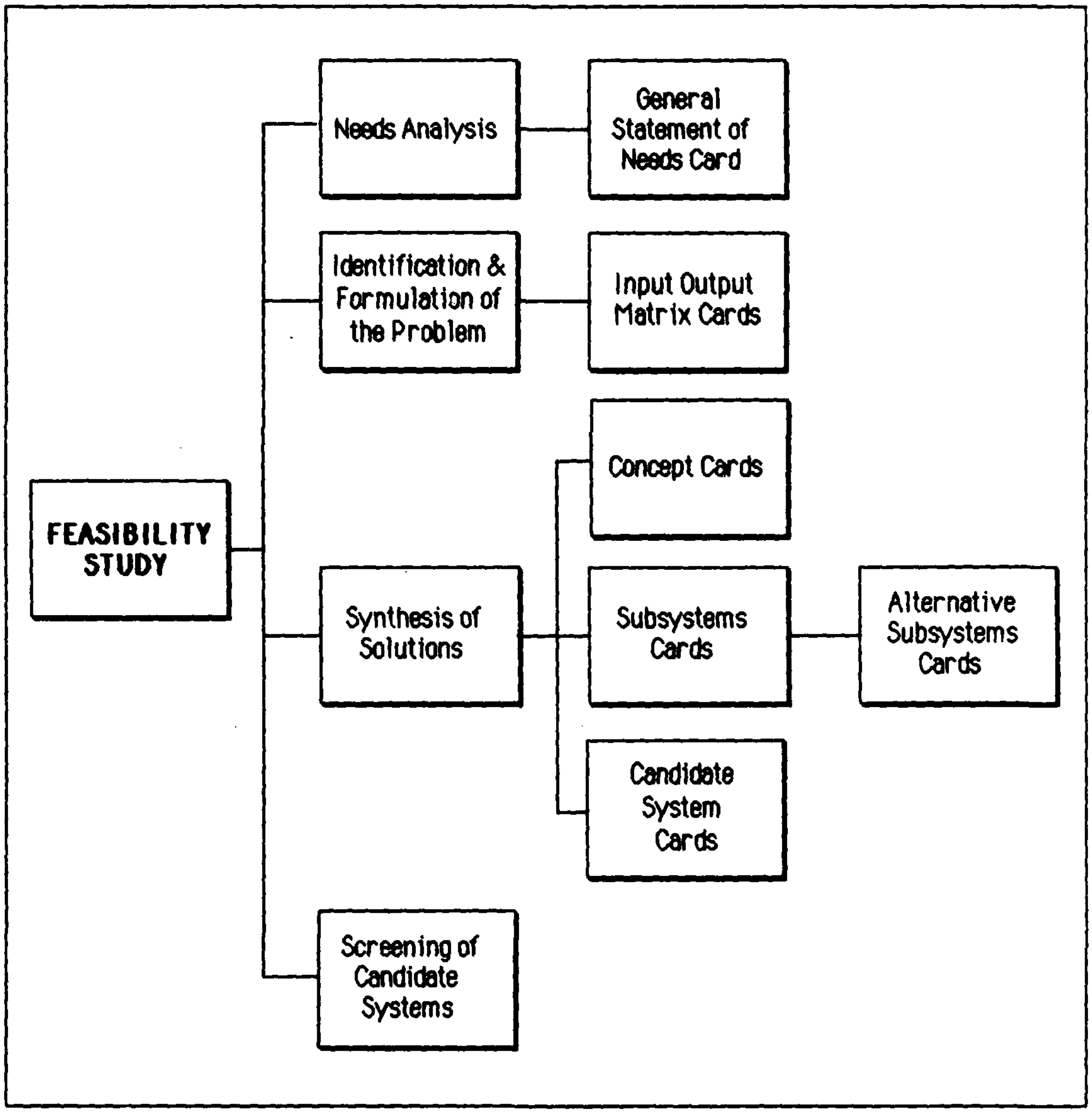


Figure 9. Initial Conceptualization

This conceptualization closely mimics the flow chart of the Feasibility Study as outlined by Ostrofsky (1977:27), shown in Figure 3 of Chapter 2.

Structuring the Stack. Danny Goodman, the foremost author on HyperCard® design and development, considers stack structuring the most important step in the stack design process (Goodman, 1988:89). Goodman suggests that a good way to begin the structuring process is to build a preliminary stack, using roughed out cards as place-holders. This stack can then be used to test ideas about how the user will navigate through the stack.

In the process of completing these activities, and while writing preliminary HyperTalk™ code to test necessary procedures, it was discovered that not all desired functions of the stack could be implemented within the HyperCard® environment. Specifically, the initial conceptualization called for drawing the concept charts on a card within the stack and allowing the user to access a card listing the alternatives for a specific subsystem by "clicking" on the representation of that subsystem. This concept was not feasible, due to limitations of the HyperCard® program. In the current version of HyperCard® the size of a card is limited to the 9 inch diagonal screen of a MacintoshPlus®, even when a larger screen is available. This was not enough room to chart even the simplest concept. Additionally, is a concept chart was drawn in the HyperCard® environment, the arrows connecting subsystems would need to be re-drawn each time a subsystem was moved. These limitations necessitated the consideration of using another medium for the actual drawing of concept charts.

Another change which was made to the initial structure during the course of the preliminary design process was the addition of the hook book as a means for filling in the Input/Output Matrix.

Because of these changes, the stack structure was modified (Figure 10) to include the hook book function and a separate "Help" stack, which is

intended to be available from all cards within the main stack. The modified stack structure also provides means for the user to access Design/IDEF™, a diagramming tool and data dictionary generator produced by Meta Software Corporation (see Appendix A).

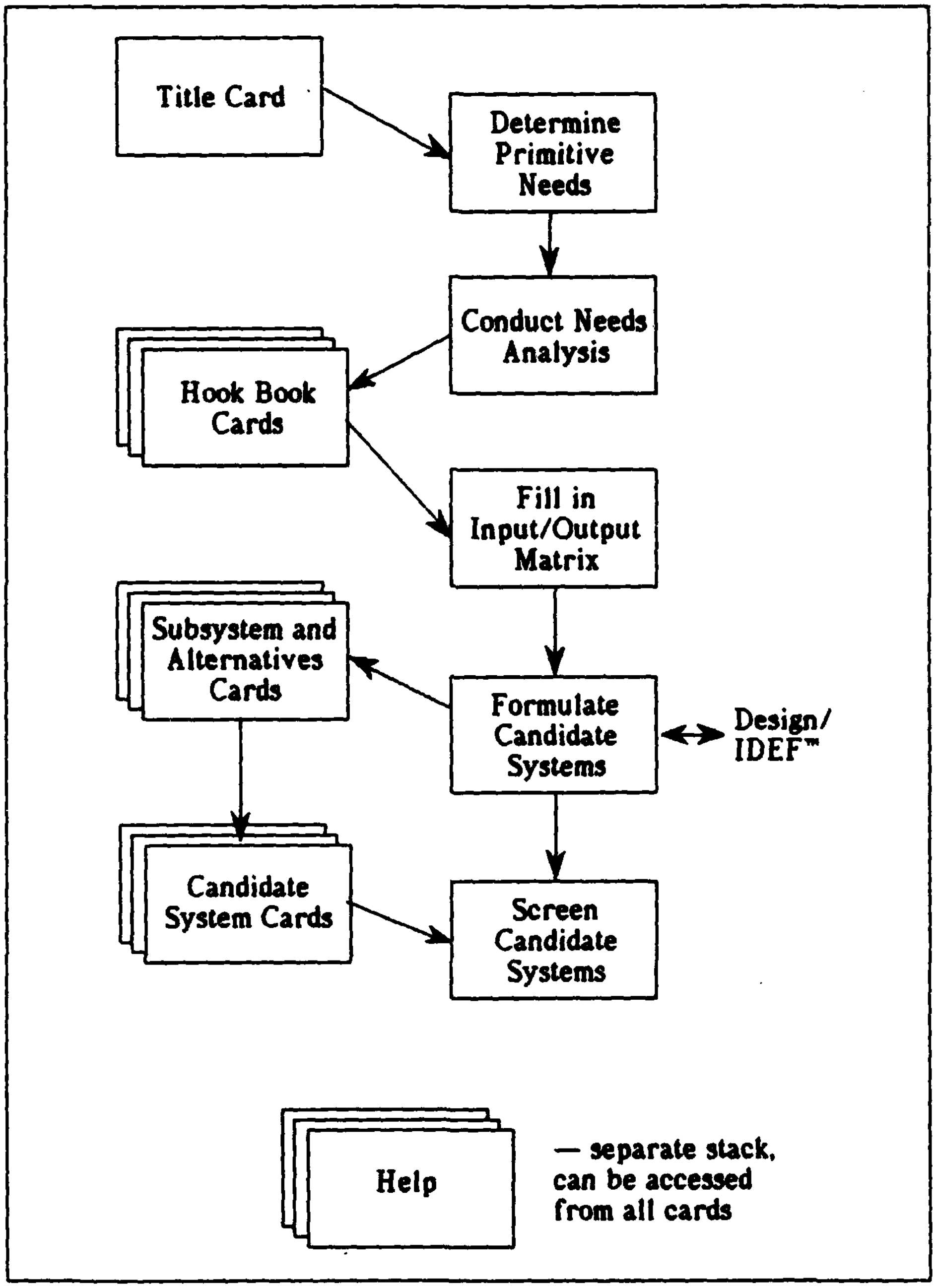


Figure 10. Modified Stack Structure

The modified stack structure takes advantage of HyperCard®'s ability to launch external application programs. When the user quits the external program, control of the computer is returned to the HyperCard® program, and the user is taken back to the card that launched the external program. The stack access into the Design/IDEF™ program allows the user to draw concept charts in an environment specifically designed for this purpose. It also takes advantage of HyperCard®'s ability to open and read documents stored on disk, and place them into a field on a card. The user will be able to bring subsystem names from the concept chart into the stack for further analysis. If the Design/IDEF™ program is not available, the user is provided with the option of manually feeding the subsystem names into the stack.

Implementing the Procedures. The stack was designed as an automation of an existing process, and all attempts were made to keep the contents of the stack as consistent as possible with the underlying *Design, Planning and Development Methodology*.

This consistency is provided by giving users the option of navigating the stack using HyperCard® adaptations of the flow charts used throughout the text. This provides a familiar environment for seasoned users, and gives beginners a way to relate the stack to the text.

Finalizing the Stack. HyperCard® relies heavily on the use of metaphor for conveying understanding, and a key part of the stack design process is figuring out how appropriate metaphors can help the user understand the stack and how to use it (Vaughn, 1988:67). During the design of the stack, all attempts were made to keep graphics consistent within the stack, with other HyperCard® stacks, and with the charts shown in the *Design, Planning and Development Methodology* text. The primary consideration made during graphic design was to ensure users familiar with the methodology would be comfortable using the stack.

General Design Considerations

Making the stack easy to use was the main consideration made during stack design. All attempts were made to make the computer and the stack as transparent as possible to the user, allowing the user to focus on the methodology, not on the medium which should be guiding him or her through the process.

Assumptions

Several assumptions were made during the design of the stack, concerning both the potential users and the necessary hardware and software.

Users. It was assumed that the potential users of the stack are familiar with the HyperCard® program and its' metaphors, and understand how to navigate through a HyperCard® stack. Users are not expected to understand the HyperTalk™ language, or produce any HyperTalk™ scripts.

It was also assumed that the potential users are familiar with the Apple Macintosh® operating system, and that explanations of the common functions of the operating system itself were not necessary.

The potential users were assumed to be familiar with, and have available for reference, the Design, Planning, and Development Methodology textbook (Ostrofsky, 1977).

Hardware. The minimum hardware requirements for using the stack are a MacintoshPlus® computer with at least one megabyte of RAM and a Hard Drive. To take full advantage of the design options available in Design/IDEF™, at least two megabytes of RAM are required.

Software. The user was assumed to have available the following applications:

1. HyperCard® (version 1.2.1 or later)
2. Reports, a report generator for HyperCard®. Reports is used to print the Input/Output matrix.

3. Design/IDEF™ (version 1.1 or later). Design/IDEF™ is used draw the concept charts, and generate subsystem listings. Manual procedures are available for users who do not have this software available.

Complete information on these programs is available in Appendix A.

IV. *The Design Support System*

> "Well, not the next day," the Knight repeated as before: "not the next day. In fact," he went on, holding his head down, and his voice getting lower and lower, "I don't believe that pudding ever was cooked! In fact, I don't believe that pudding ever will be cooked! And yet it was a very clever pudding to invent."
>
> — Lewis Carroll (1872)

The Design Support System which resulted from the research activities described in Chapter Three consists of three elements, the Main Stack, referred to as the "Design, Planning and Development Stack", the User's Manual, and the Help Stack. Each of these elements is described in terms of how they are viewed and used by the IRM system designer.

The Design, Planning and Development Stack

Figure 11 is a logical map of the Design, Planning and Development (D, P & D) Stack. The cards in the stack can be categorized as navigational, action, or data, according to the primary purpose of the card. Navigational cards help the user get from one point in the stack to another. Action cards carry out specific IRM system design activities. Data cards collect the data and information generated by these activities. This is a general categorization made for the purpose of simplifying explanation; all cards contain some navigational aspects, and most of the action cards have some data collection aspects.

The main path through the stack, from Title Card to Screening of Candidates Card (shown by the bold line in Figure 11), implements the Feasibility Study as outlined in Ostrofsky's *Design, Planning and Development Methodology* (1977:28) (refer to Figure 3, Chapter 2). The iterative nature of the *Design, Planning and Development Methodology* is supported by the multiple paths depicted in Figure 11, and by the capability to backtrack along any of these paths.

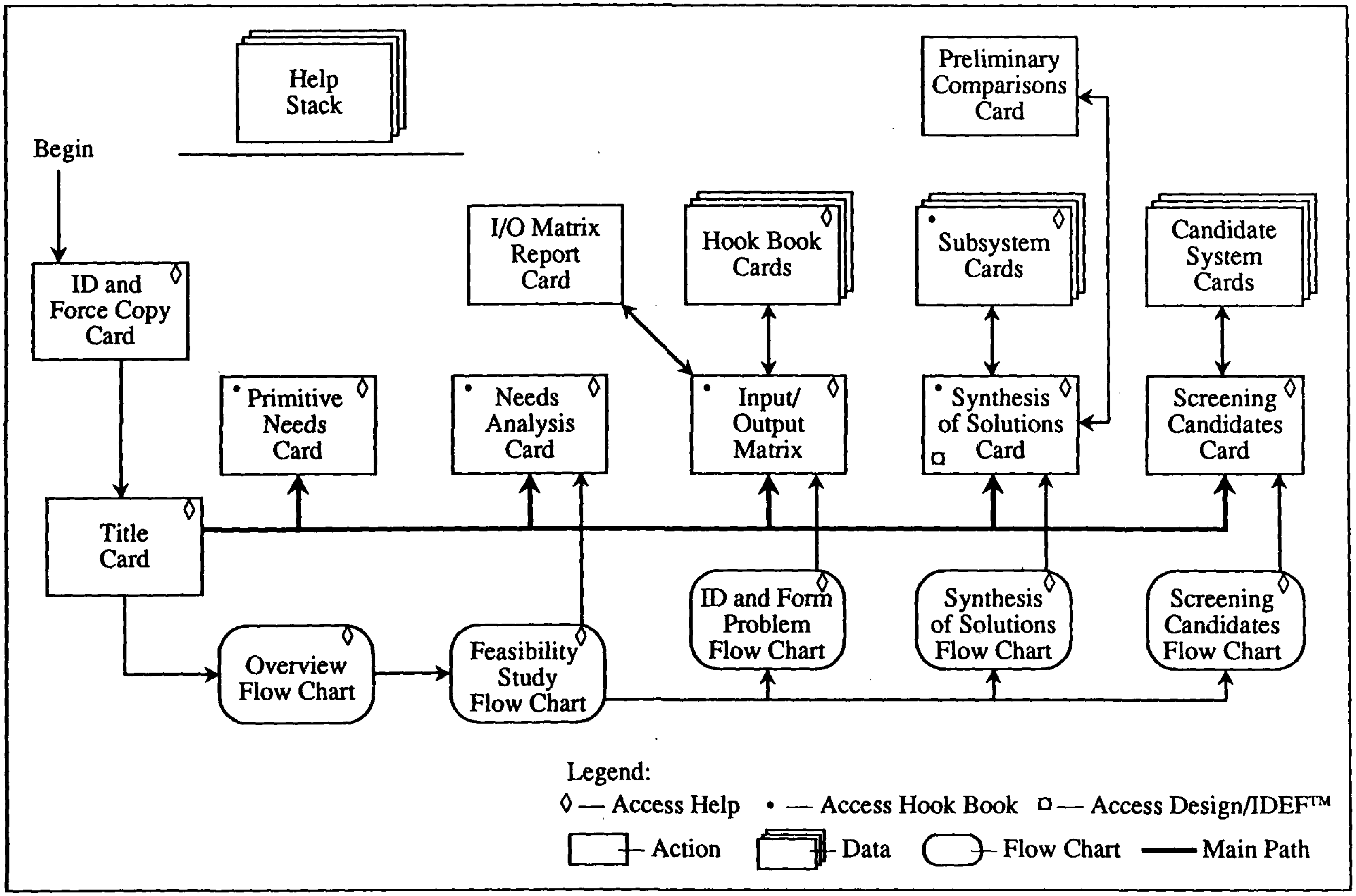


Figure 11. D, P & D Stack Logical Map

Navigation in the Stack. The user has two primary means for navigating through the stack: the Flow Chart Index and the Action Index.

Flow Chart Index. The Flow Chart Index is made up of cards containing flow charts identical to those pictured throughout the Design, Planning and Development textbook (Ostrofsky, 1977). An example Flow Chart Index card is shown in Figure 12.

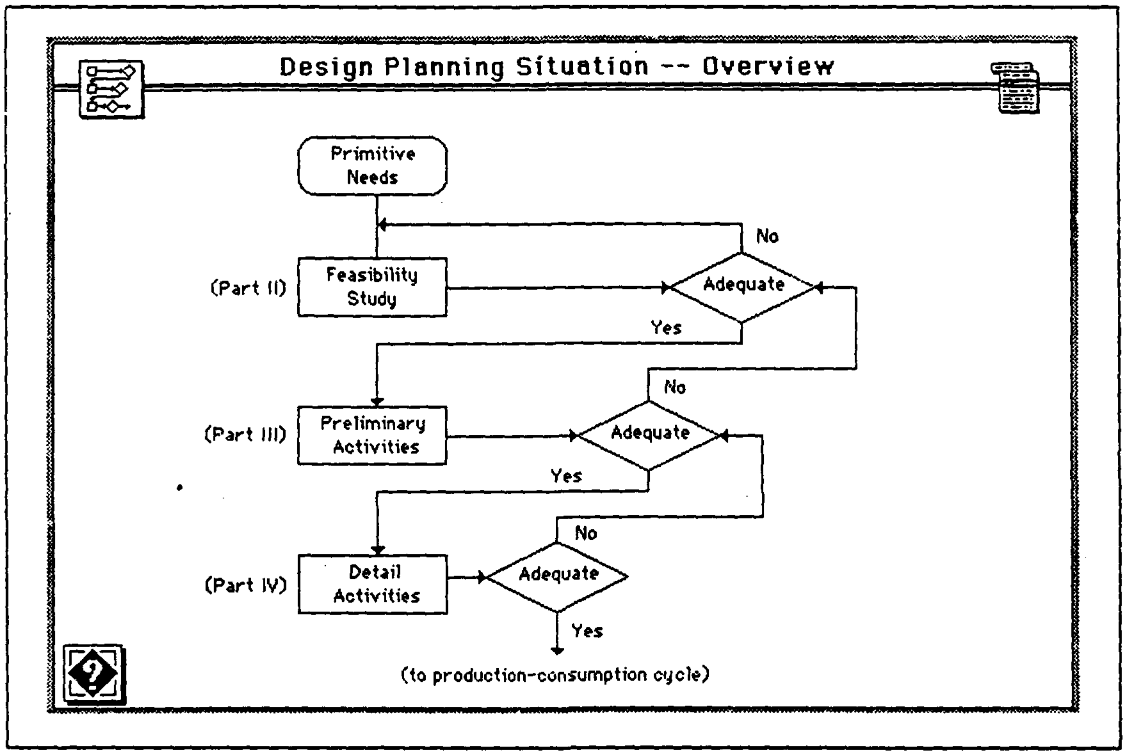


Figure 12. Example Flow Chart Index Card

The flow charts lead the user familiar with the methodology through the D, P & D stack in a recognizable manner, and help ensure he or she remains oriented. This second aspect is extremely important; one of the primary complaints voiced about hypertext is that it is very easy for the user to become disoriented, or "lost in hyperspace" (Conklin, 1987:57). The flow charts provide a map for the user, and allow the user to back out of an action and return to familiar ground to re-orient.

Symbols on the flow charts are actually buttons. When a user clicks the mouse on a symbol, he or she is either taken further down the flow chart hierarchy to another flow chart, or is presented with a "pop-up" informational note (Figure 13).

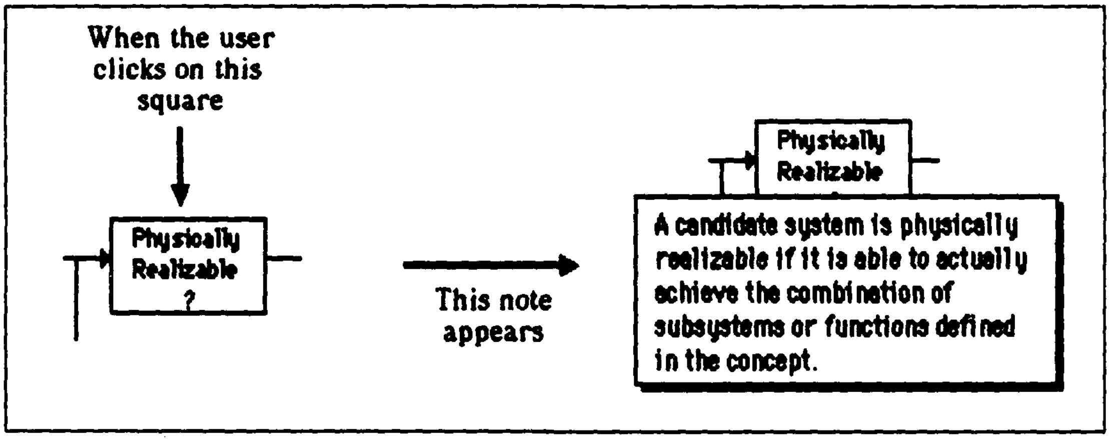


Figure 13. Example of a Pop-Up Informational Note

Action Index. The Action Index is accessed by the button shown in Figure 14. This button appears on the Flow Chart Index cards, the Action

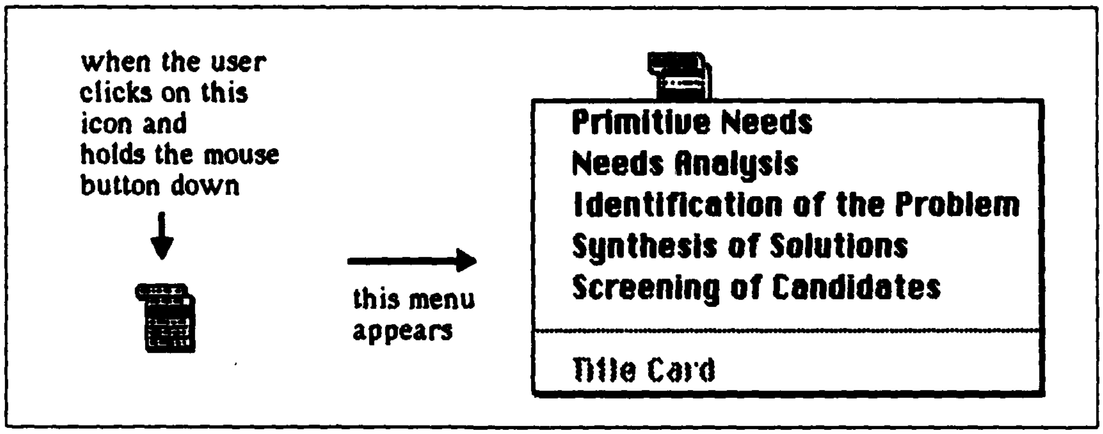


Figure 14. Action Index, Viewed From the Title Card

cards, and the Title Card. It allows the user to directly access the Title Card or any of the five Action cards. If the user is viewing the Title Card or

an Action card, the title of the card that he or she is viewing will be dimmed. This helps orient the user by providing a reminder of where he or she is along the main path.

Other Navigational Aids. Figure 15 shows the buttons which are used to navigate throughout the stack.

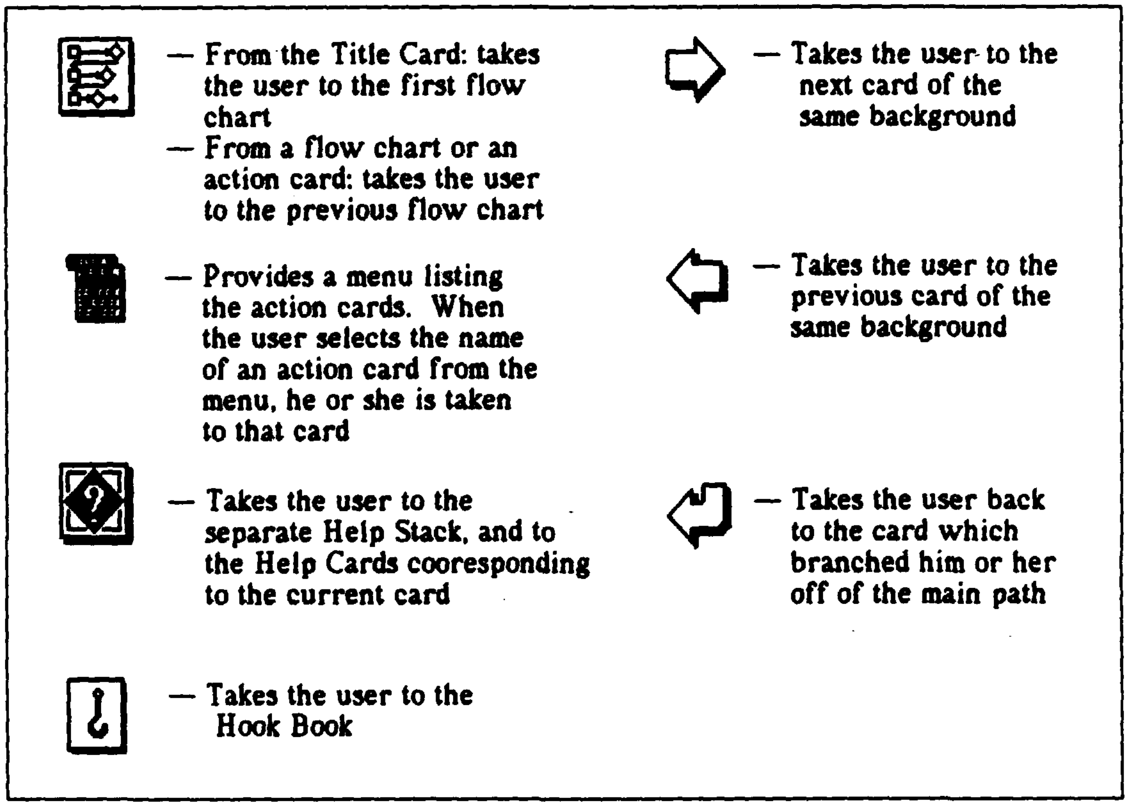


Figure 15. Navigational Buttons

Description of the Cards. Upon opening the D, P & D Stack, the first card the user encounters is the Identification and Force Copy Card (Figure 16).

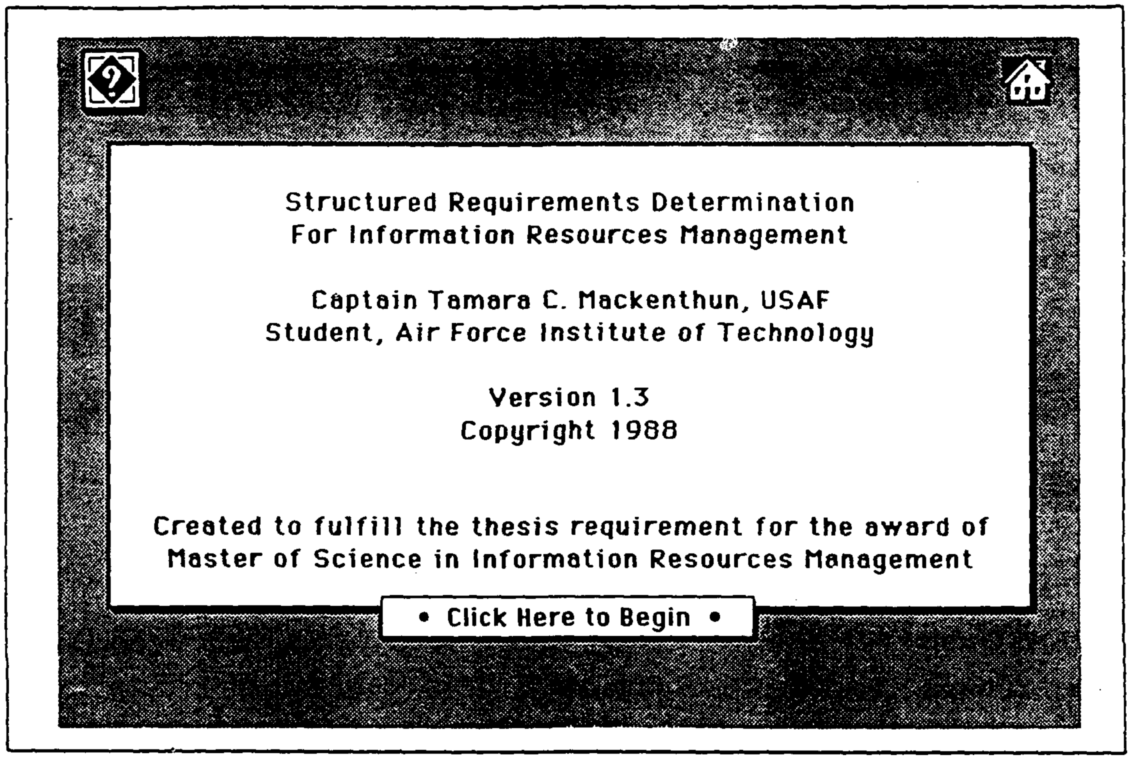


Figure 16. Identification and Force Copy Card

This card forces the user into a loop which can only be terminated by leaving the stack or making a copy of the stack. This capability forces the user to make a working copy, specific to the design problem at hand, thus ensuring data integrity. The working copy will not contain this card, but will begin with the Title Card.

The **Title Card** is the primary orientation point for the user (Figure 17).

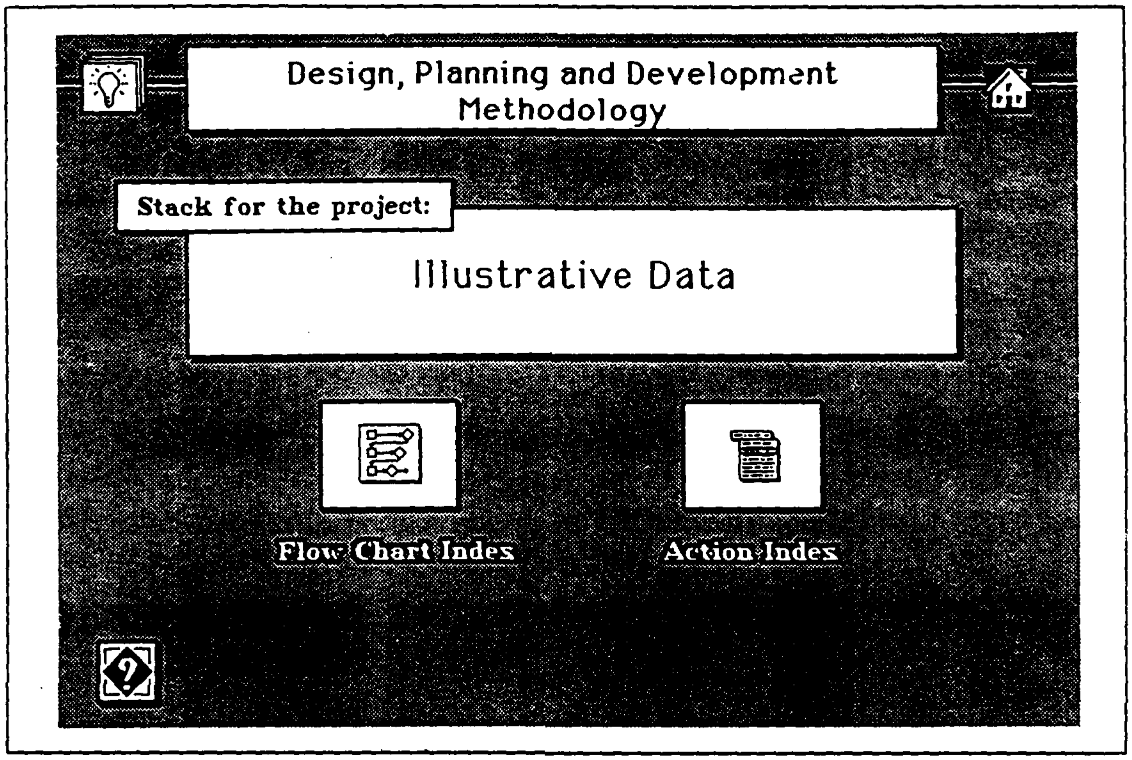


Figure 17. Title Card

The Title Card contains the name of the project, access to the Flow Chart Index, Action Index, and Help Stack. The Title card contains two additional buttons which are not commonly used throughout the stack, the Lightbulb button in the top left corner, and the Home button in the top right corner. The Lightbulb button presents the user with the stack information identical to that displayed on the Identification and Force Copy Card. The Home button allows the user to exit the stack and go to HyperCard®'s visual directory, the "Home Stack." Going to the Home Stack allows the user to navigate to other HyperCard® stacks or exit the HyperCard® environment.

From the Title Card the user navigates, either by means of the Flow Chart Index or the Action Index, down the main path, completing each action in the Feasibility Study along the way. The user is not forced to

remain on this path; he or she can always go back to a previous card, or jump ahead to a later card, in accordance with the iterative nature of the underlying methodology.

The first Action Card along the main path is the **Primitive Needs Card** (Figure 18).

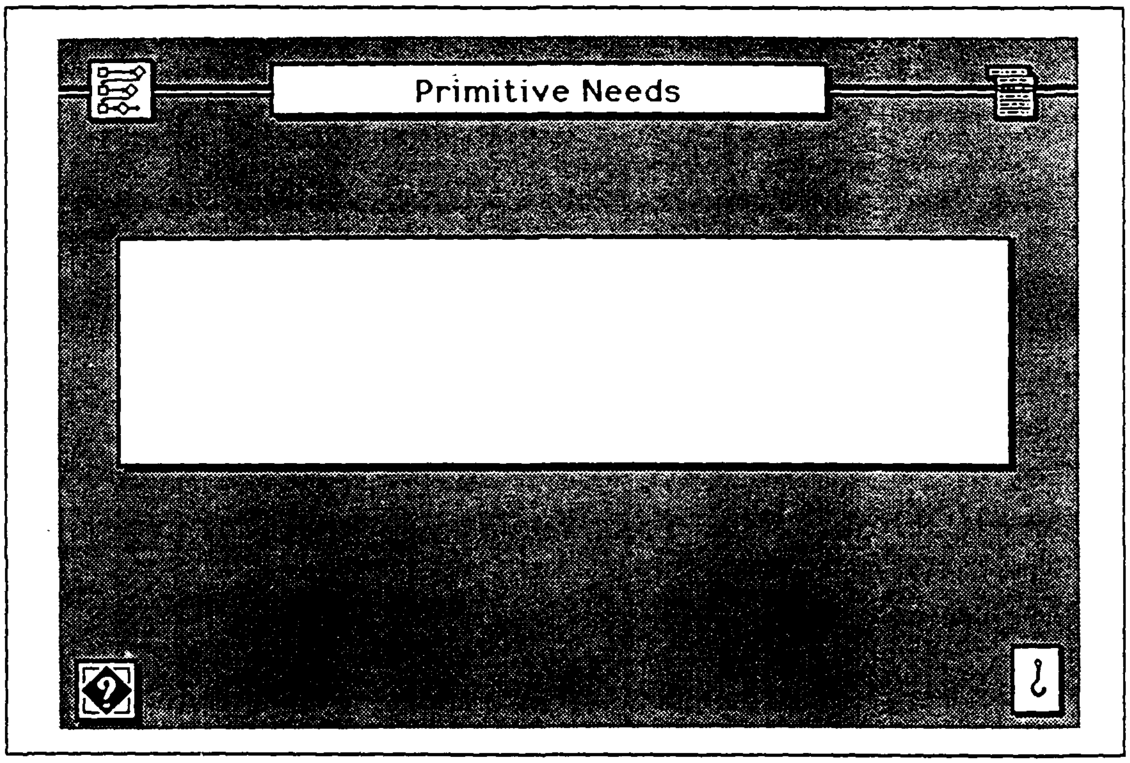


Figure 18. Primitive Needs Card

This card provides a place for the user to identify and store the description of the Primitive Needs, the one sentence problem statement which focuses the entire design process. The user can freely access and/or change the Primitive Needs statement should that be required.

The next action card along the main path is the **Needs Analysis Card** (Figure 19).

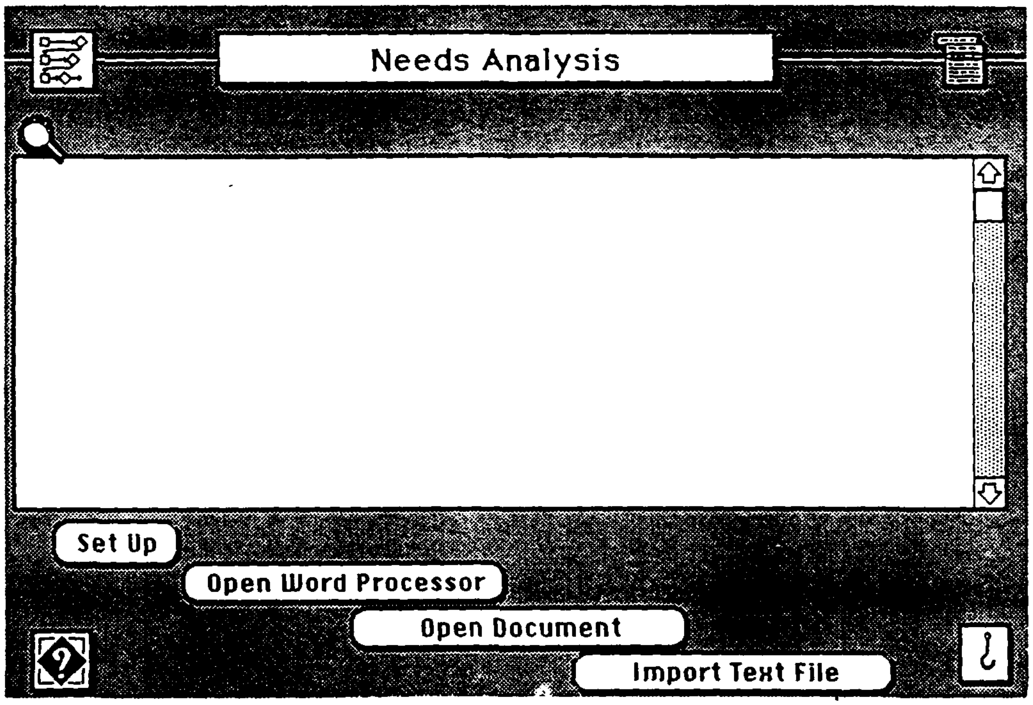


Figure 19. Needs Analysis Card

The end product of the Needs Analysis is a statement of the needs which the resulting system must satisfy, and a justification for further expenditure of resources. Because this product is primarily a text document, this card allows the user to launch a word processing program, write the needs analysis, save it as a word processing document for inclusion in other reports, and import the text of the analysis into the stack for easy reference during the design process. The four buttons at the bottom of the Needs Analysis card allow the user to specify what word processing program he or she will be using, launch the program, directly access the Needs Analysis document, and import the document into the field on the Needs Analysis card. The button which looks like a small magnifying

glass (at the upper left corner of the text field) allows the user to search through the imported text for a specific word or phrase.

After completing the Needs Analysis, the user moves down the main path to the next activity, the Identification and Formulation of the Problem, carried out using the **Input/Output Matrix Card** (Figure 20).

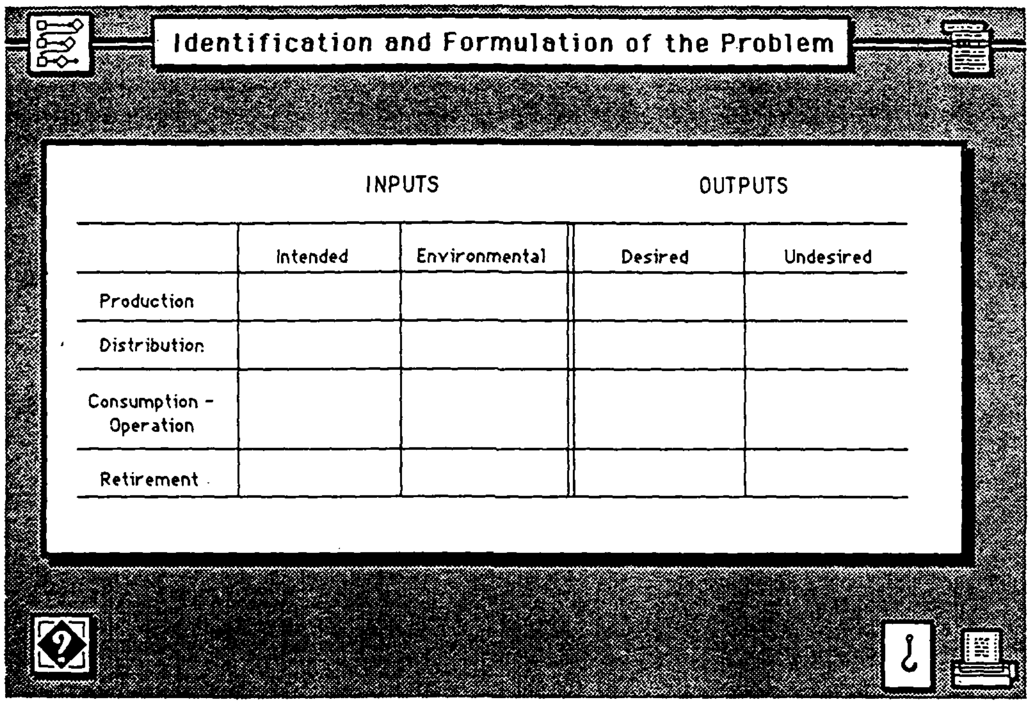


Figure 20. Input/Output Matrix Card

The Input/Output Matrix Card is the user's main interface with the **Hook Book Entry Cards** (Figure 21).

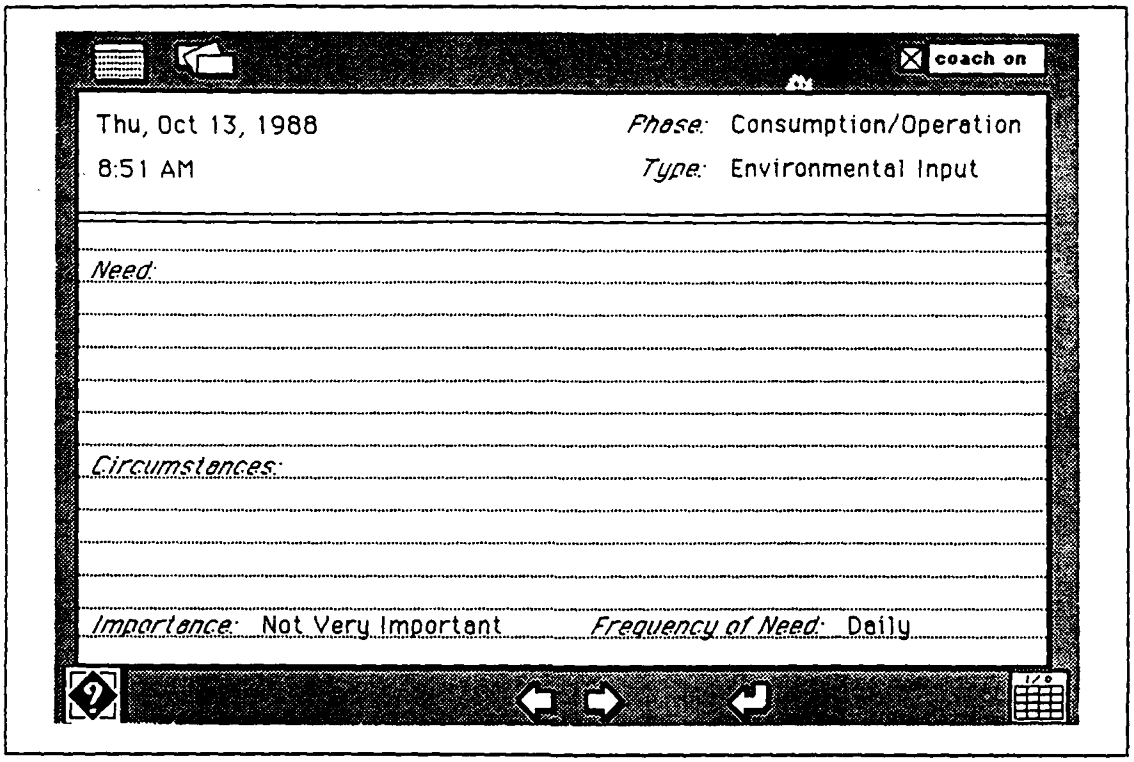


Figure 21. Hook Book Entry Card

The Hook Book Entry Cards can be accessed from five points in the stack (see Figure 11); however, all Hook Book Entry Cards feed their data into the Input/Output Matrix Card. Hook Book Entry Cards are the user's means for "jotting down" what they want the resultant IRM system to accomplish. The Hook Book Entry Cards also provide a place for the user to document their prioritization of needs.

After the user completes a Hook Book entry, he or she can feed that entry into the Input/Output Matrix, using the "commit" option from the Entry Button Menu, as shown in Figure 22.

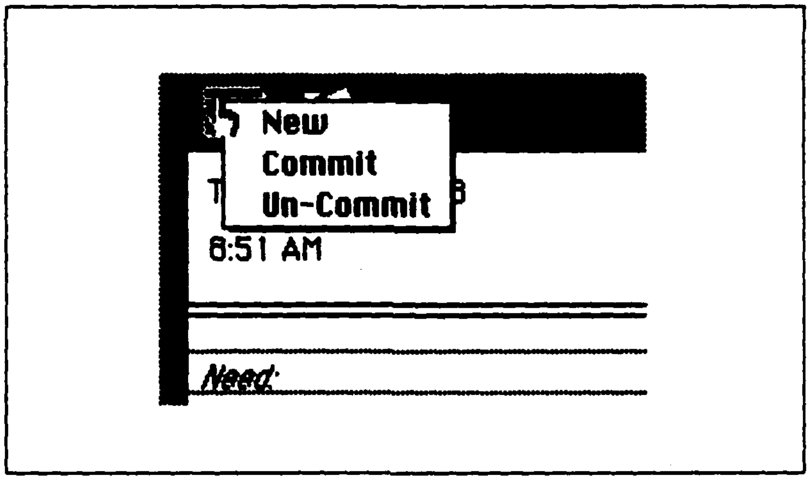


Figure 22. Hook Book Entry Button Menu

The entry is then fed into the appropriate box on the Input/Output Matrix Card. This box will be "checked" to show that it contains at least one entry (Figure 23).

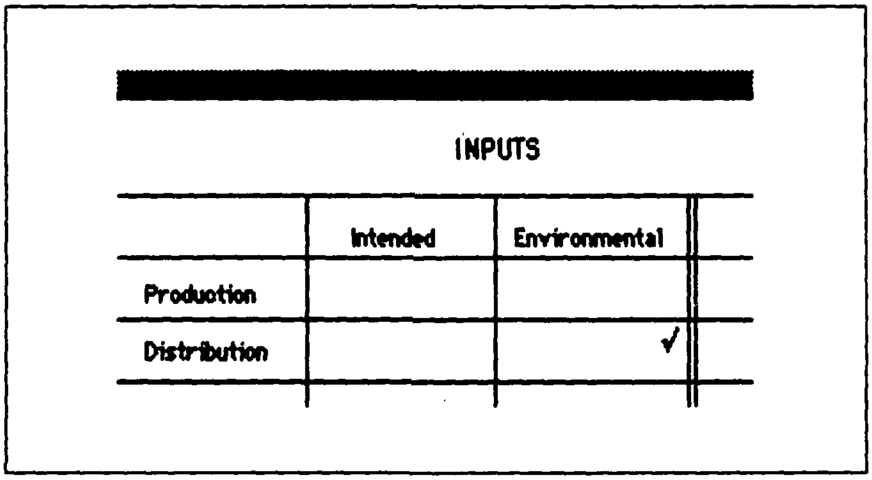


Figure 23. Portion of the Input/Output Matrix Card, Annotated to Reflect a Committed Entry

The user can view entries by clicking the mouse on a box, which causes that box to "zoom" open (Figure 24).

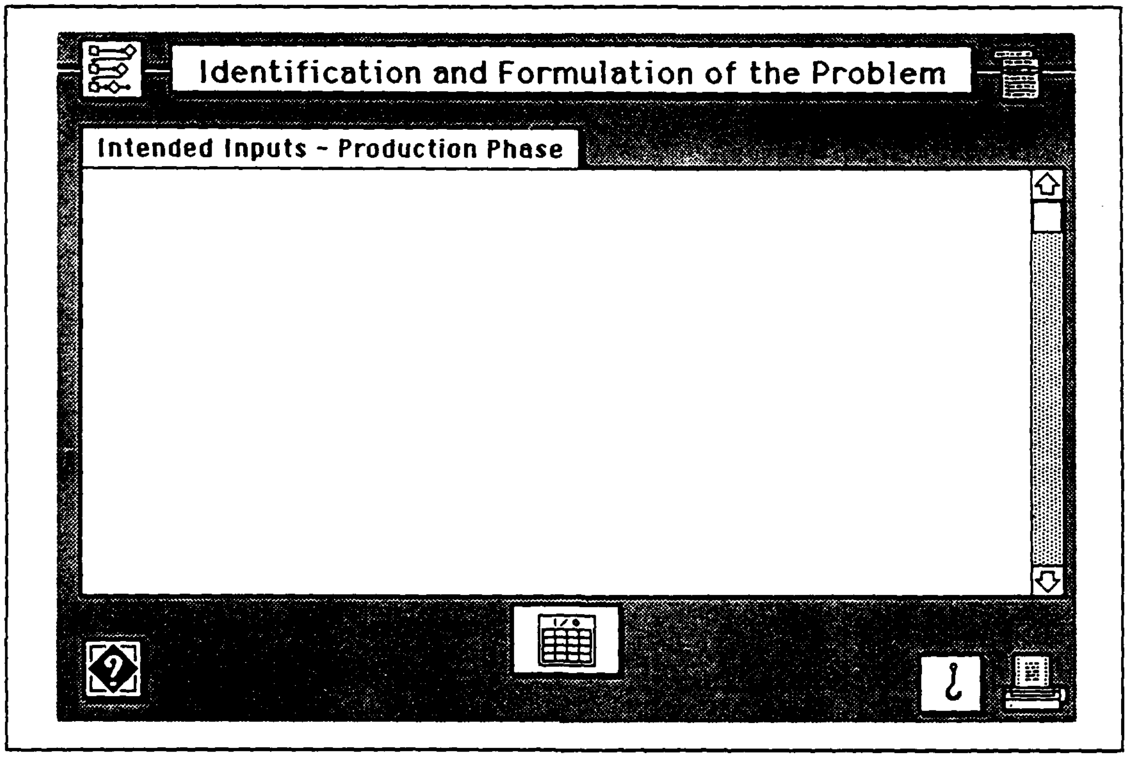


Figure 24. Input/Output Matrix With a Box "Zoomed" Open

The user can print the entries committed to the Input/Output Matrix by clicking on the printer button in the bottom right corner of the Input/Output Matrix Card. This button launches the Reports program, and uses the report formats supplied with the D, P & D Stack.

The next action card along the user's main path is the **Synthesis of Solutions Card**. Figure 25 shows this card with one of the three concept menus visible.

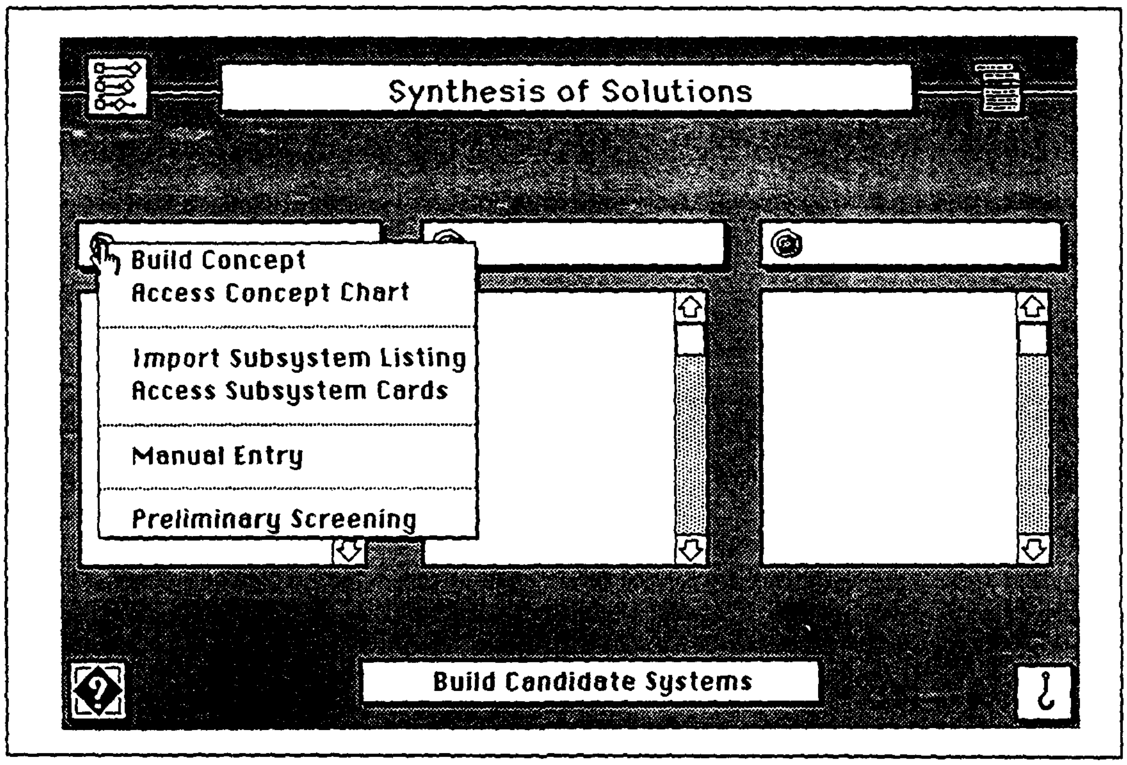


Figure 25. Synthesis of Solutions Card

These menu options allow the user to:

• Construct graphic representations of three different concepts, referred to as Concept Charts, using the Design/IDEF™ program.

• Read the subsystem names in from a text file created in Design/IDEF™ and create **Subsystem Cards** (Figure 26).

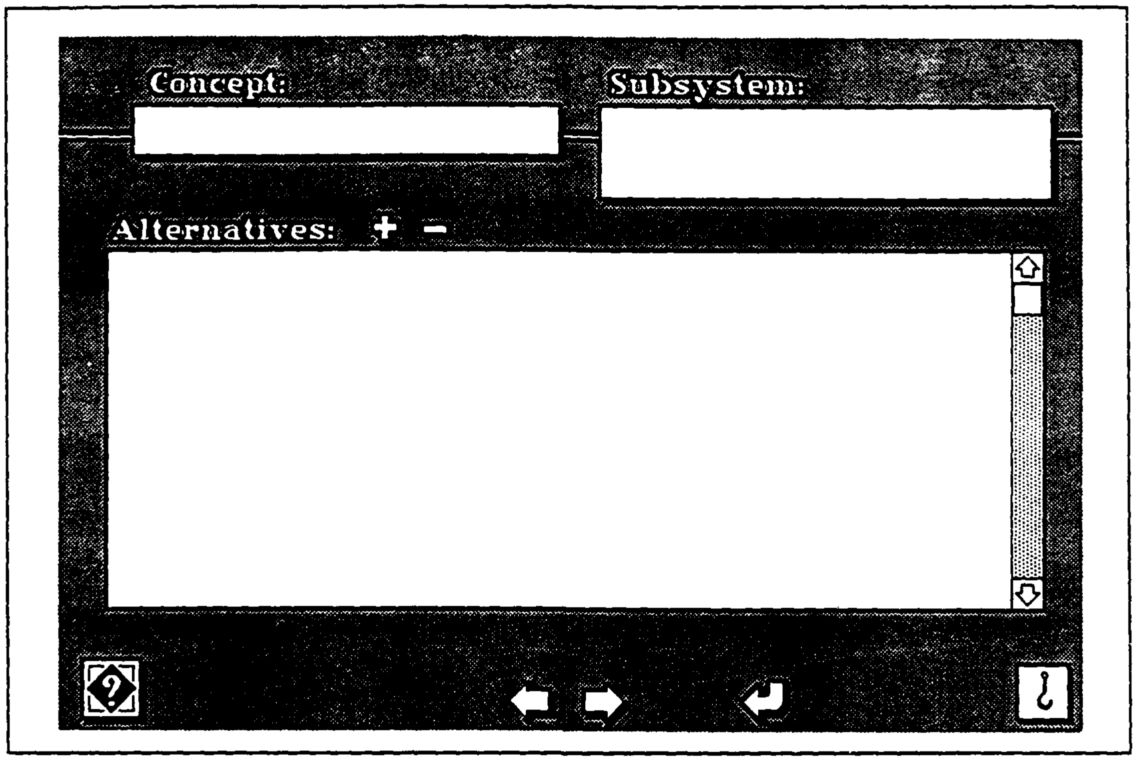


Figure 26. Subsystem Card

• Manually enter the subsystem names if the Design/IDEF™ program is not available.

• Access existing Subsystem Cards where alternatives can be added or deleted using the " + " and "-" buttons located directly above the "Alternatives" field.

• Conduct a preliminary screening of alternatives to avoid creating candidate systems containing subsystems which are clearly incompatible.

The button at the bottom of the Synthesis of Solutions Card builds the actual candidate systems.

After building the candidate systems, the user moves on to the next card along the main path, the **Screening of Candidate Systems Card** (Figure 27).

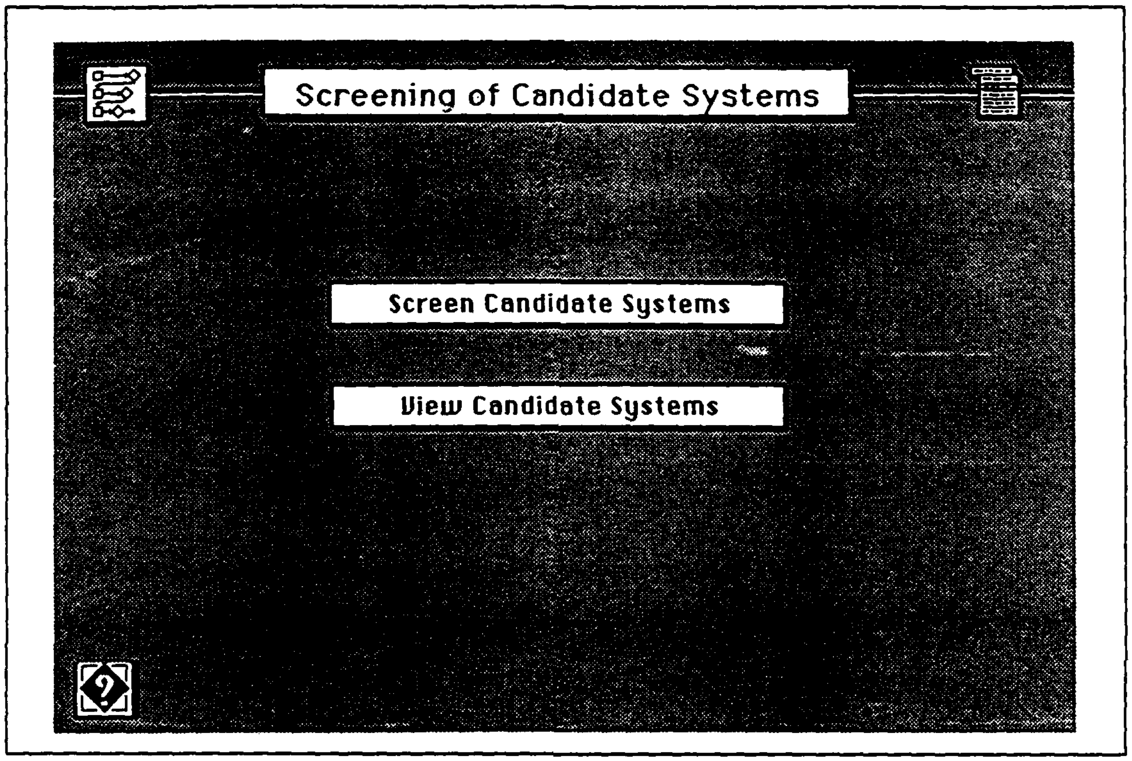


Figure 27. Screening of Candidate Systems Card

This card leads the user through the process of screening the candidate systems to ensure they are physically realizable, economically worthwhile and financially feasible. The user eliminates those candidate systems that are clearly impossible to develop.

The user can also view the **Candidate System Cards** (Figure 28) from the Screening of Candidate Systems Card.

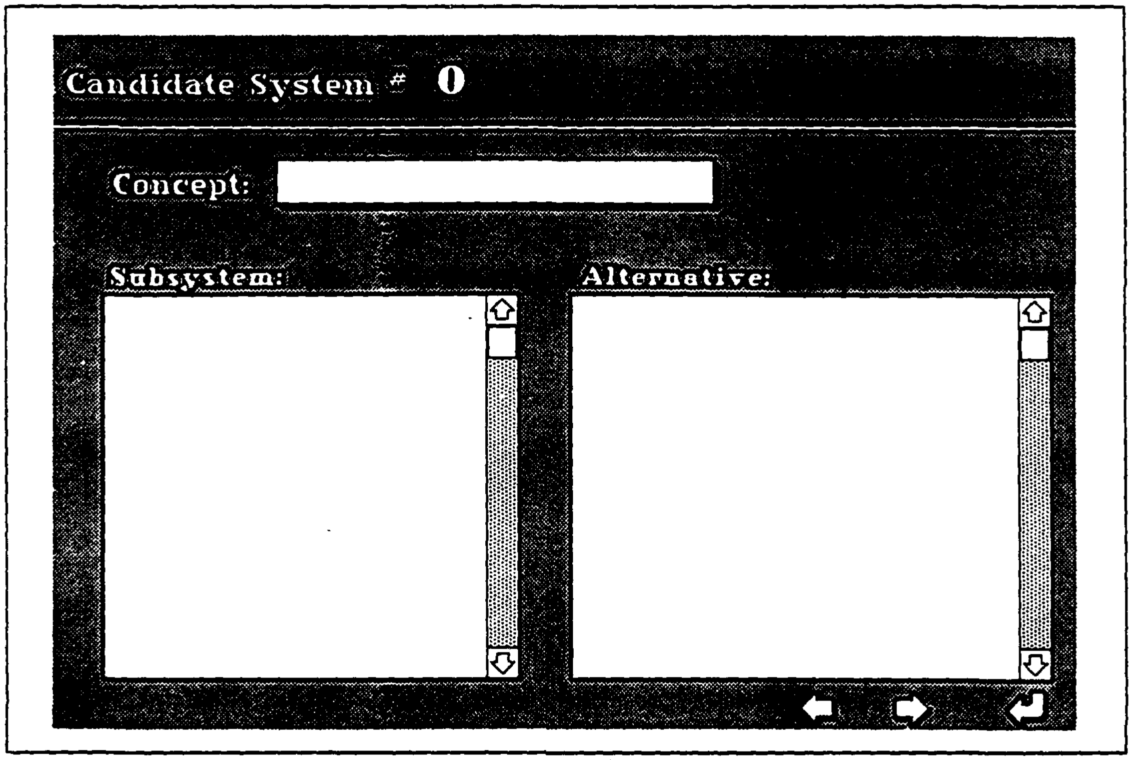


Figure 28. Candidate System Card

Upon reaching this point, the user has completed all necessary tasks in the Feasibility Study. He or she now has a set of feasible candidate systems which can be analyzed using the second design planning phase in the *Design, Planning and Development Methodology*, the Preliminary Activities.

Help Stack

The Help Stack is a separate stack which can be accessed from almost any point in the D, P & D Stack. The cards in the Help Stack explain the purpose of a particular card, how to use the specific buttons on that card, and how to navigate through the stack. When a user clicks on a Help

Button, he or she is taken into the Help Stack, to the first Help Card which corresponds to the card from which he or she left the D, P & D Stack.

The User's Manual

The User's Manual provides a general description of the system and its purpose, and provides information not included in the Help Stack. The User's Manual consists of four sections, and is contained in Appendix B.

Section I describes the hardware and software requirements, the background the user should have, and a list of suggested references.

Section II shows the user how to copy the automated portions of the system onto a hard disk, and how to initially access the D, P & D Stack. It also describes the structure of the D, P & D Stack, and explains how to access the Help stack.

Section III describes Concept Mapping and explains how it can be used to start a Concept Chart.

Section IV describes the Design/IDEF™ program and how it can be used to draw a concept chart which is compatible with the D, P & D Stack.

V. *Conclusions and Recommendations*

> **...when they had been running half an hour or so, and were quite dry again, the Dodo suddenly called out "The race is over!" and they all crowded round it, panting, and asking "But who has won?"**
>
> **— Lewis Carroll (1865)**

Conclusions

The stated objective of this research was to develop a system to support the decisions which must be made by the Information Resources Management System designer.

Ostrofsky's *Design, Planning and Development Methodology* was chosen as the decision making model to apply to this problem. This methodology provides a framework upon which the IRM system designer can structure the decisions which must be made in order to translate rapidly changing needs into plans which implement rapidly changing technology.

Apple Computer's HyperCard® programming environment and Meta Software's Design/IDEF™ diagramming and data dictionary program were used to develop the Design, Planning and Development (D, P & D) Stack.

The D, P & D stack, combined with its' accompanying help stack and user's manual, make up a design support system which guides the IRM system designer through the requirements determination phase of the *Design, Planning and Development Methodology*, assists in the collection of data, and organizes that data into a form which can be subjected to objective analysis and optimization.

Recommendations for Further Research

The Stack Itself. The D, P & D Stack is far from complete. There are a number of areas which warrant further research, namely:

• Building the candidate systems. The actual computer instructions for building the candidate systems are not included in the stack. This omission is the result of a recognition that even the generation of candidate systems for the small number of alternatives identified in an elementary sample problem would cause the stack to become so large that it would be impossibly slow. Including the actual definition of candidate systems will necessitate a re-design of the stack into a system of stacks.

• On-Line Help. As it exists, the help function built into the D, P & D stack is incomplete. Most of the Help Cards contain only a fundamental explanation of how to step through the actions in the D, P & D stack, there is little discussion of the underlying methodology, or why a specific step needs to be taken. The Help function should be expanded to include tutorial items on the methodology itself. It should also include IRM system specific examples.

• HyperCard® Card Size. As discussed in Chapter 3, the initial conceptualization of the stack did not require the user to access an external program for the generation of the concept charts; however this idea had to be abandoned because of the limited size of a HyperCard® card. It is possible that the HyperCard® program will be enhanced to allow the creation of variable sized cards. If this occurs, the stack could be redesigned to allow the user to carry out all design functions within the stack itself. This would also allow the concept chart to play a more active role in navigation and organization; the representation of a subsystem in a concept chart could be an active button which accesses the corresponding subsystem card.

• Expand to include the remainder of the methodology. The D, P & D Stack only supports the Feasibility Study, or requirements determination phase of the *Design, Planning and Development Methodology.* The stack

needs to be expanded into a complete system which provides support for all phases of the design process.

The Complete Design Support System

• The steps outlined in Section III of the User's Manual for using concept mapping techniques to help design the concept charts have not been tested or validated. This is an area rich in research possibilities.

• The Design Support System has not yet been implemented in a real world problem situation. The system needs to be used in an actual design situation in order to identify weaknesses or problem areas.

HyperCard® (version1.2.1)

Apple Computer, Inc.
2052 Mariani Ave.
Cupertino, California 95014
(408) 996-1010

Reports

Activision, Inc.
P.O. Box 7287
Mountain View, California 94039
(415) 329-7699

Design/IDEF™ (version 1.1)

Meta Software Corporation
150 Cambridge Park Drive
Cambridge, Massachusetts 02140
(617) 576-6920

The D, P & D Stack

a copy of the D, P & D Stack can be obtained from:

Lt Col Richard Peschke
AFIT/LS
Wright Patterson AFB, Ohio 45433
(513) 255-4437
Autovon 785-4437

The Design, Planning and Development Stack

User's Manual

I. The Stack

Purpose

The Design, Planning and Development (D, P & D) Stack is intended to guide the Information Resources Mangement System designer through the requirements determination process using the principles outlined in Ostrofsky's *Design, Planning and Development Methodology*.

Hardware Required

The minimum system configuration for the D, P & D Stack is an Apple Macintosh® computer, equipped with at least one megabyte of RAM and a hard disk drive.
Two megabytes of RAM are recommended.

Software Required

You will need the following software in order to use the D, P & D Stack:

- **HyperCard®**, version 1.2 or later produced by Apple Computer Inc.
- **Reports**, a report generator for HyperCard, produced by Activision Inc.
- **Design/IDEF™**, a diagramming tool and data dictionary, produced by Meta Software Corporation*

Complete vendor information can be found in the references at the end of this section.

These applications should be loaded to your hard disk according to the instructions provided in their respective manuals.

* Provisions are made for those users who do not have the Design/IDEF progam available.

User Experience

• You should be familiar with the Macintosh® operating system.

If you have never used a Macintosh computer before, please take some time to become familiar with the operation of the computer itself.

• You should have some experience using HyperCard®

If you are not familiar with HyperCard®, please go through the tutorial provided with the program.

• You should be familiar with Ostrofsky's *Design, Planning and Development Methodology*

There are many terms and concepts used in the D, P & D Stack which are specific to this underlying methodology. If you have never worked with or studied the methodology, you should read through the first section of The *Design, Planning and Development Methodology* (Ostrofsky, 1977)

Suggested References

The Operating System:

Kaehler, Carol. *Macintosh ® Plus User's Manual.* Cupertino CA: Apple Computer, Inc., 1986.

Apple Computer, Inc. *Technical Introduction to the Macintosh Family.* Reading MA: Addison-Wesley Publishing Company, Inc., 1987.

The Software:

Apple Computer Inc. *HyperCard ® User's Guide.* Cupertino CA: Apple Computer Inc., 1987.

Goodman, Danny. *The Complete HyperCard™ Handbook.* New York: Bantam Books, 1988.

Vaughn, Tay. *Using HyperCard ®: From Home to HyperTalk™*. Carmel IN: Que Corporation, 1988.

Snow, Janice and Randall Albright. *Design/IDEF™ User's Manual.* Cambridge MA: Meta Software Corporation, 1987.

The Methodology:

Ostrofsky, Benjamin. *Design, Planning and Development Methodology.* Englewood Cliffs NJ: Prentice Hall Inc., 1977.

Concept Mapping:

Novak, Joseph D. and D. Bob Gowin. *Learning How to Learn.* New York: Cambridge University Press, 1984.

Software

HyperCard ® (version 1.2.1)

Apple Computer, Inc.
2052 Mariani Ave.
Cupertino, California 95014
(408) 996-1010

Reports

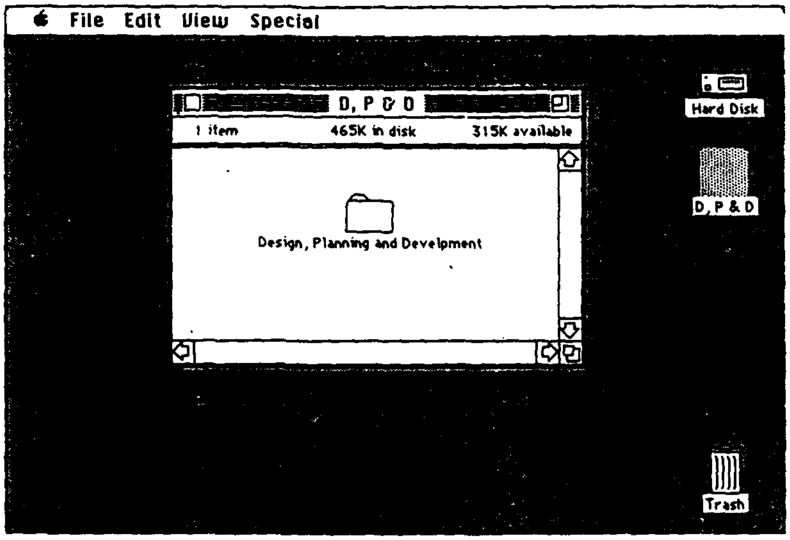
Activision, Inc.
P.O. Box 7287
Mountain View, California 94039
(415) 329-7699

Design/IDEF™ (version 1.1)

Meta Software Corporation
150 Cambridge Park Drive
Cambridge, Massachusetts 02140
(617) 576-6920

Copying to a Hard Disk

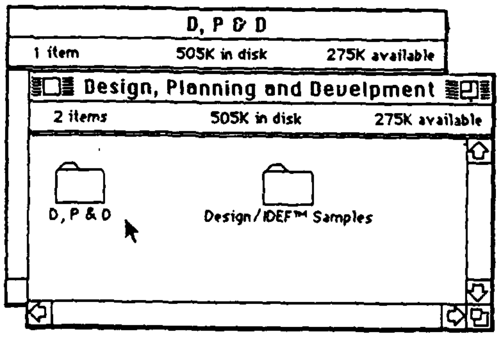
When you open the D, P & D disk, you will see one folder, entitled "Design, Planning and Development."



- Copy this folder onto your hard disk
- Close this window and
- Eject the D, P & D disk

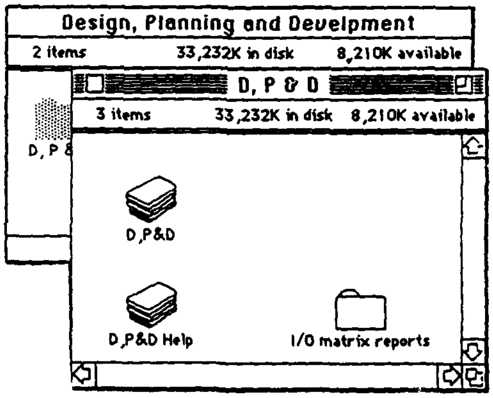
To begin using the D, P & D Stack, open the Design, Planning and Development folder which you copied to your hard disk.

You will see two folders:

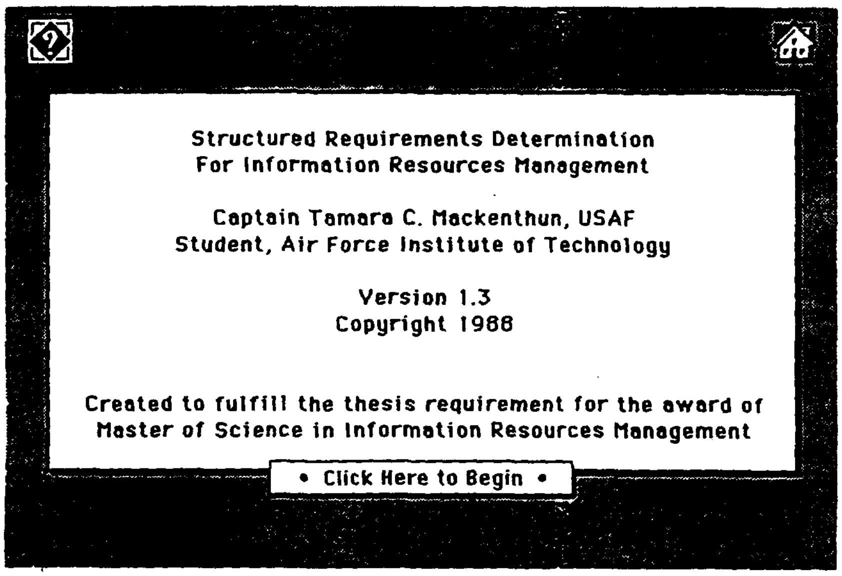


• Open the D, P & D folder

• Click on the stack titled "D,P&D"

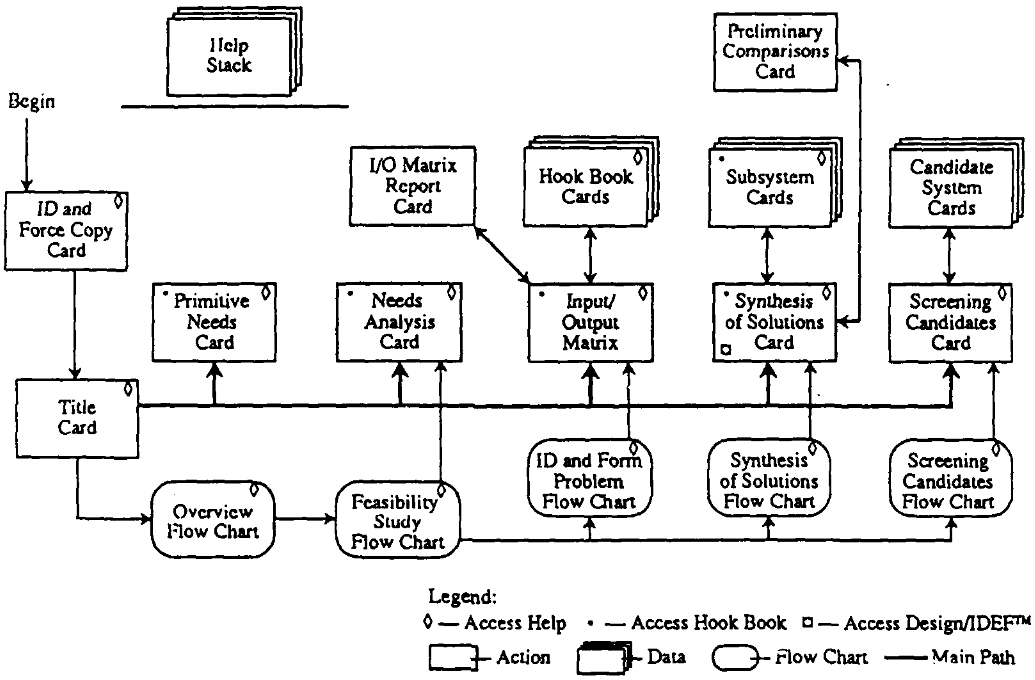


You will now be viewing the first card of the D, P & D Stack.



Stack Structure

The chart below is a logical map of the D, P & D Stack.



The main path through the stack (shown with the bold line) will take you to the Action Cards, where you will carry out the requirements determination activities of the Feasibility Study.

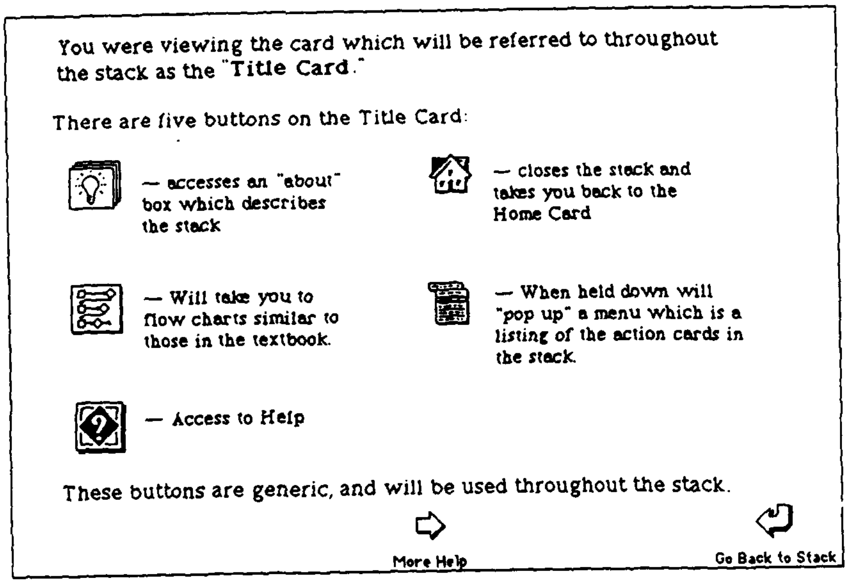
Help Stack

If at anytime while using the D, P & D Stack you do not understand how to proceed,

- Click on the **Help Button**: 

You will be taken to the D, P & D Help Stack, where you will have access to a set of **Help Cards** corresponding to the card you were viewing in the D, P & D Stack.

The figure below is a sample Help Card, accessed from the Title Card:



Help Stack Buttons

There are four navigational buttons used throughout the Help Stack:

More Help

— Takes you to the next Help Card on this subject

Previous Card

— Takes you to the previous Help Card

View Help Again

— Takes you to the first Help Card on this subject

Go Back to Stack

— Takes you back to the card in the D, P & D Stack where you clicked on the Help button

III. Designing a Concept Chart

Concept Chart

The Concept Chart is a visual representation of the subsystems which make up a concept and how they relate to each other.

The purpose of the concept chart is to structure concepts, or basic approaches to the solution of the design-planning problem. In order to draw a concept chart, the designer must first translate the needs of the production-consumption cycle (identified in the input/output matrix) into the elemental activities, or subsystems, which will meet these needs.

This can be a difficult process. Essentially, the designer needs to identify the functions, tasks, and attributes of the resulting system and describe how they are related.

Some insight into the problem can be gained by using the Concept Mapping techniques described by Novak and Gowin. Concept Mapping is a tool which can be used to capture and relate the key aspects of a problem. The following specific guidelines have taken these techniques and placed them within the specific context of the *Design, Planning and Development Methodology* .

Specific Guidlines

1. Translate each need identified in the Input/ Output Matrix into a one or two word statement — either an object or an event. Do not include verbs or action statements.

2. Rank order these statements by degree of generality — list the most general statement first,

and work through the list until all statements are rank ordered by degree of generality.

3. Write each statement on a 3X5 card (or post-it note). Place the most general statement at the top, and work down and out, building a hierarchical tree.

It is recommended that you not carry out this step using a computerized drawing program, at this point in the conceptualization process the emphasis should be placed on the process itself, not on making the chart look good.

4. Link the statements with verbs or short action phrases taken from the context of the I/O matrix

5. Look for cross links between statements in one part of the tree and statements in other "branches". Link these associated statements with verbs or short verb phrases.

6. Rebuild this structure until you are comfortable with it.

7. Look for natural groupings of the statements

8. Translate the groupings into
   a. activities accomplished by the user
   b. tasks accomplished by the future system
   c. attributes of the future system

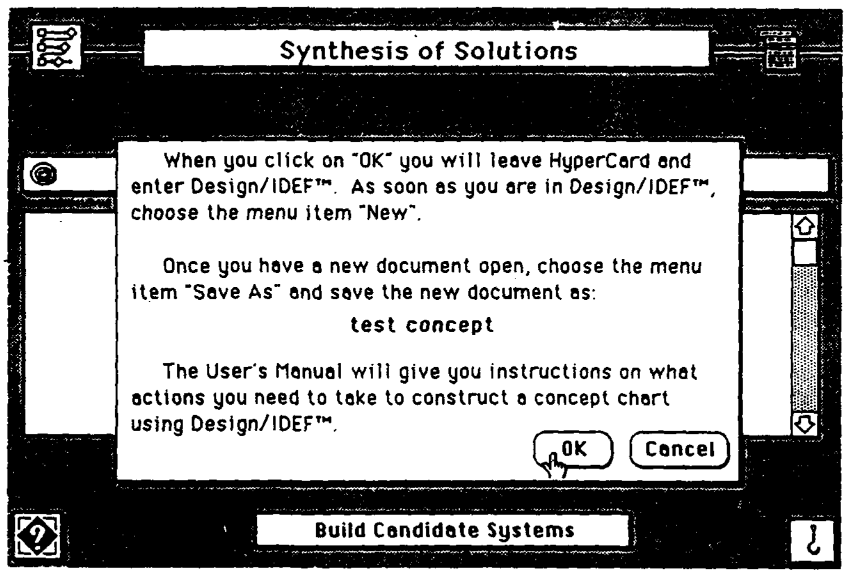
These activities, tasks and attributes are the subsystems.

9. Identify the relationships between the subsystems

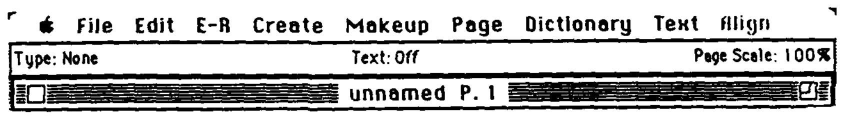
IV. Drawing a Concept Chart

This description of the Design/IDEF Program is only intended to give you an idea of how to set up the program so that the information it produces will be compatible with the processes in the D, P & D Stack. It is not intended to be an exhaustive explanation of the many diagramming tools available in the program. For a complete description of Design/IDEF's capabilities and functions, please refer to the Design/IDEF user's manual.

The description of how to build a concept chart using Design/IDEF will begin at the point where you clicked on "OK" in this box:



You should be viewing a blank screen, with the following menu items available:



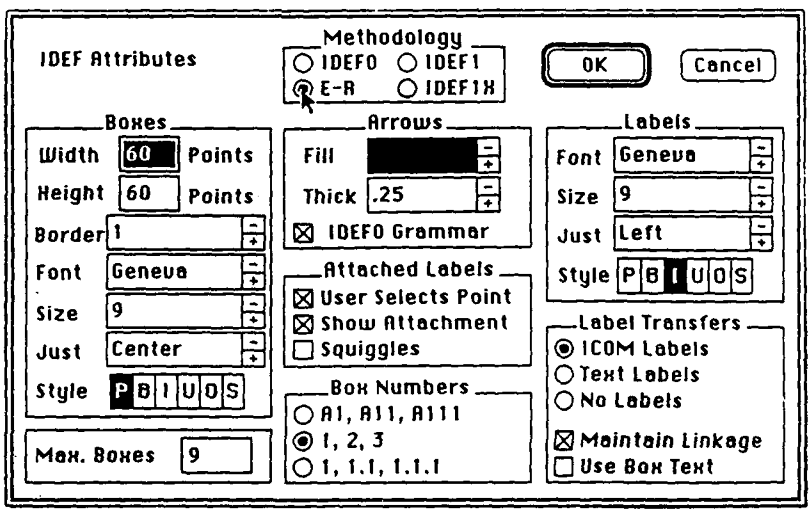
Set Up

- Select "New" from the File Menu

If the third menu title from the left is not "E-R", then:

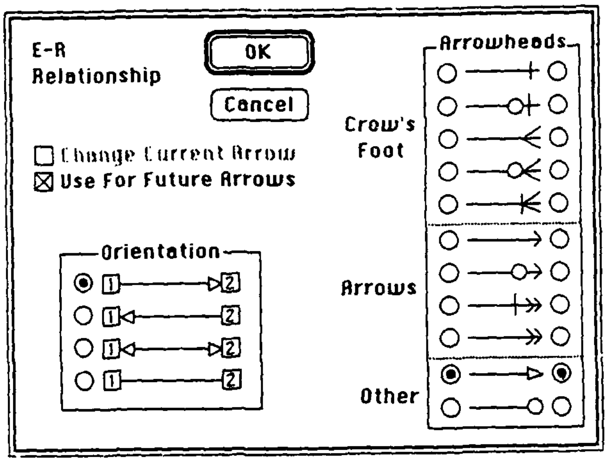
- Select "Attributes" from the third menu

- Click on "E-R" as shown



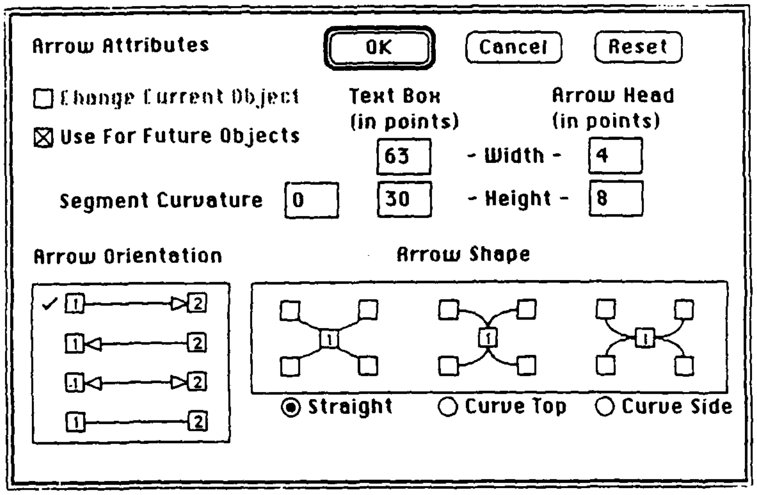
• Select "Relationship" from the E-R Menu

Choose the following arrow configurations:



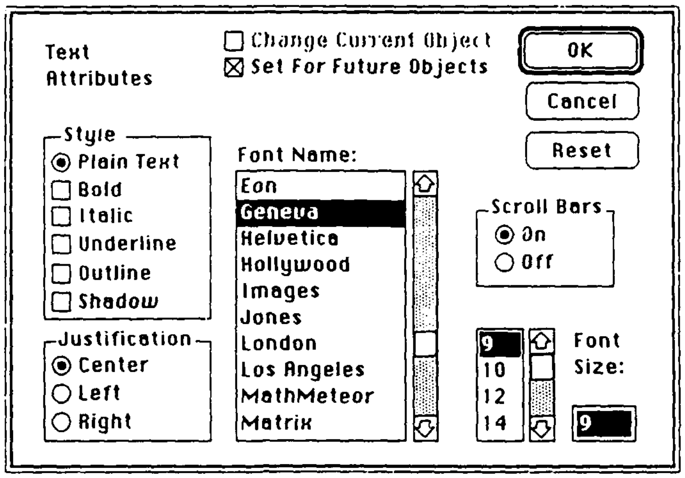
• Select "Arrow Attributes" from the Create Menu

Choose the following arrow attributes:



• Select "Attributes" from the Text Menu

Choose the following text attributes:

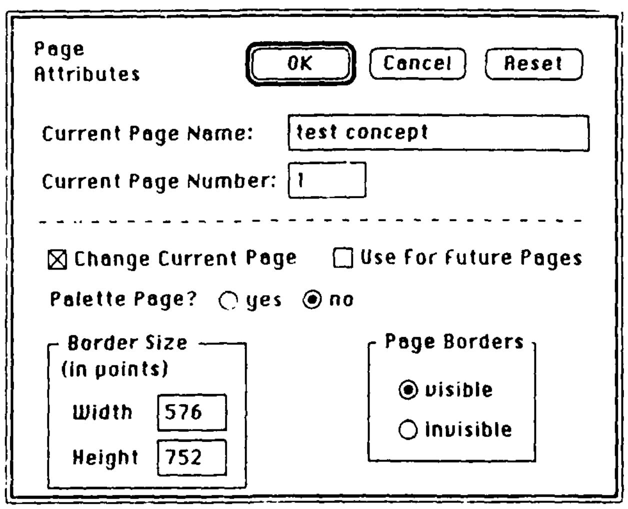


• Select "Save As" from the File Menu

Save the document using the name you declared before you left the D, P & D Stack

• Select "Attributes" from the Page Menu

Save the page using the same name:



Design

You are now ready to begin drawing your Concept Chart.

Subsystems

- Select "Turn On" from the Text Menu
- Select "Rounded Box" from the Create Menu

Place the cursor at the point on the page where you want to position your first subsystem. Click the mouse. A box containing a text insertion point will appear. Type the name of the subsystem. Repeat this process until you have drawn a box for each subsystem.

In order to move or change the size of a box:

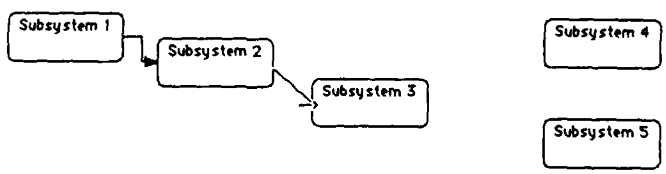
- Select "Turn Off" from the Text Menu
- Select the appropriate command from the Makeup Menu

Arrows

- Select "Turn Off" from the Text Menu
- Select "Arrow" from the Create Menu

The cursor will change to a small arrow.

Place the cursor just inside the first subsystem, click the mouse, drag the arrow to the next subsystem, place the arrowhead just inside the box and release the mouse.



Define

Now that you have finished drawing your concept chart, you need to create a data dictionary for the concept in order to be able to import the subsystem names into the D, P & D Stack.

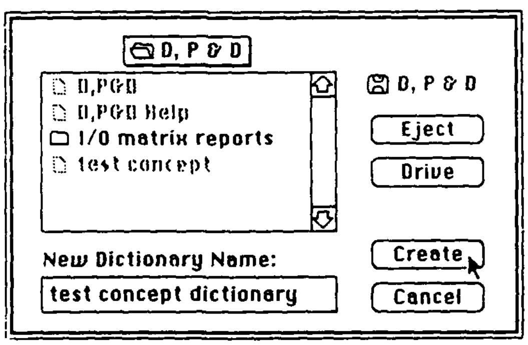
Each dictionary you create will be saved in three files, which will be placed in the same folder as your concept chart. They will be named first with the name you assign to the dictionary, and then with the *dd_idx*, *dd_dat*, and *dd_inf* suffixes. You don't need to know a whole lot about these files other than to not delete them as superfluous.

Create the Dictionary

- Choose "Select Dictionary" from the Dictionary Menu

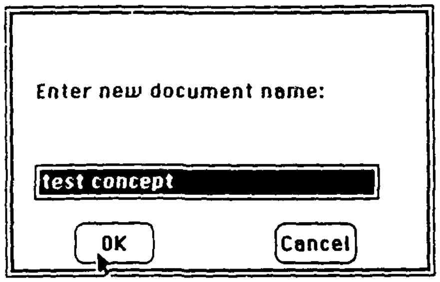
Click on the "New" button in the standard open dialog box.

This will bring up the dictionary "Save as... Dialog box. Type in the same name that you gave the document and the page, adding "dictionary" and click on "Create"



The document naming dialog box will appear, with the name of the concept chart as the default value.

Click on "OK"



Define Record Types

Every object reference in the data dictionary must be assigned a record type before it can be named. Most of the parameters which are defined in this process are not pertinent to the D, P & D problem.

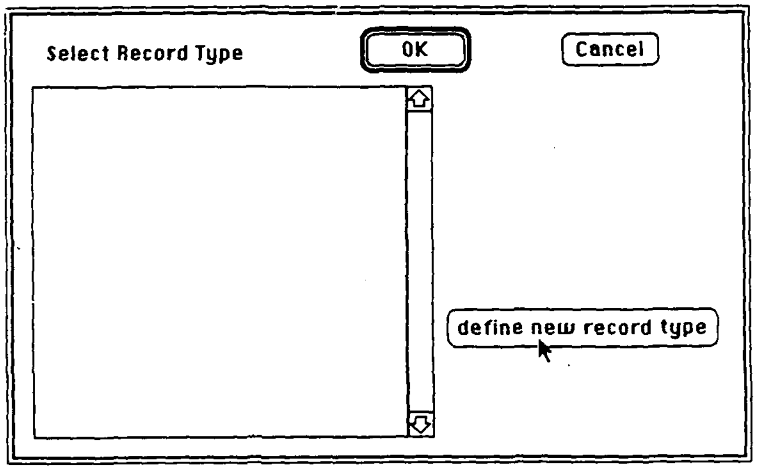
During the next few steps you will essentially be approving default values.

- Choose "Select" from the MakeUp Menu

- Click the mouse on the first subsystem

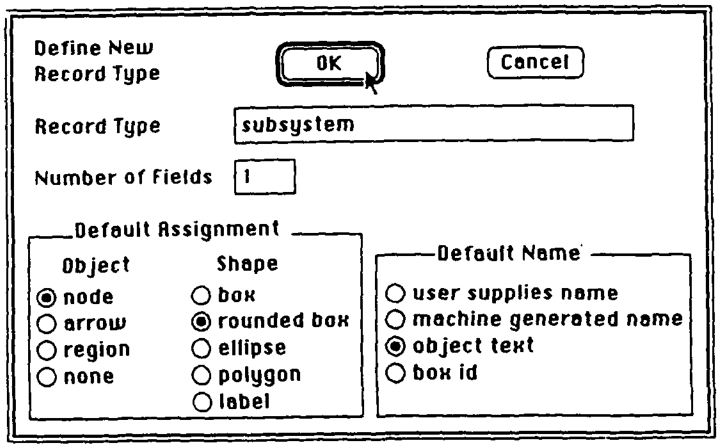
- Select "Create Object Record" from the Dictionary Menu

Click on "Define New Record Type" in this dialog box:

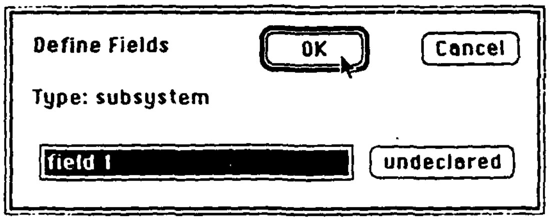


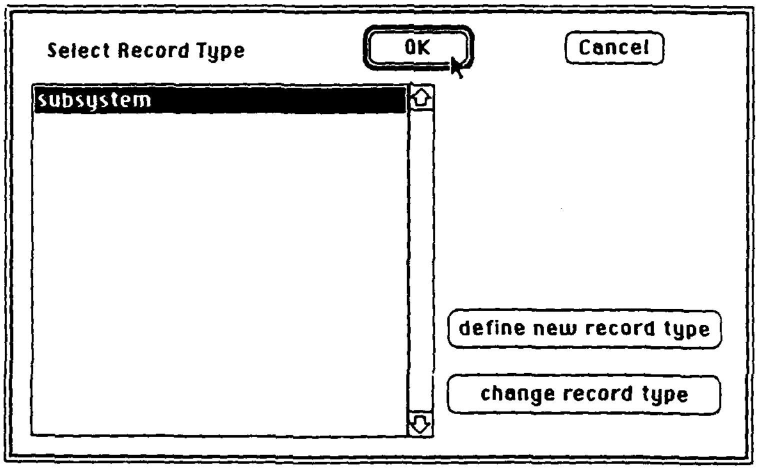
The Define New Record Type Dialog Box will appear.

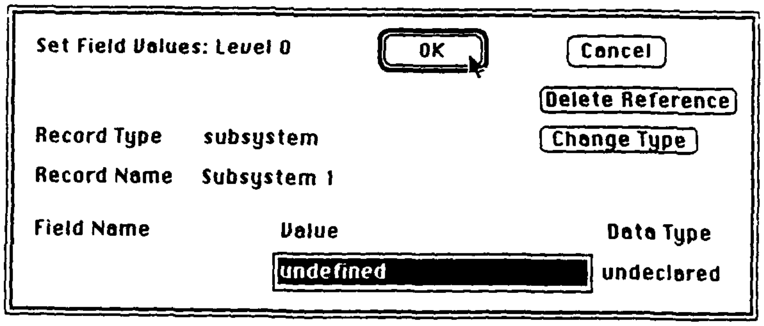
- Type "subsystem" in the Record Type box
- Select Node and Rounded Box as default Assignments
- Select Object Text as the default Name



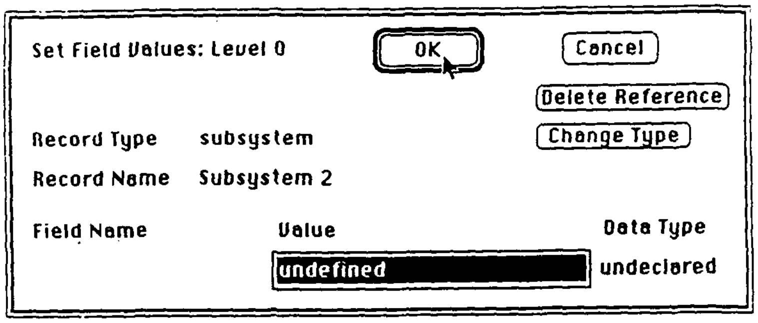
- Click "OK" in the next three dialog boxes presented, as shown







- Click on the next subsystem
- Select "Create Object Record" from the Dictionary Menu
- Click on "OK" in the dialog box



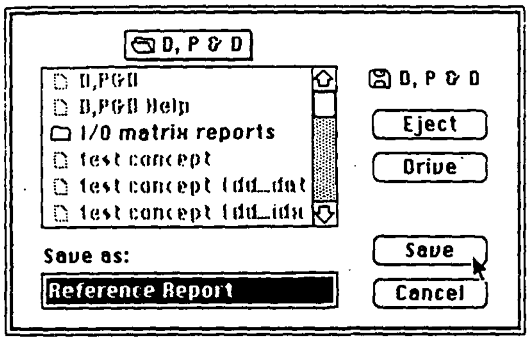
- Repeat this process for each subsystem

Create Reference Report

The Reference Report is the document which actually contains the names of the subsytems. It is the document which the D, P & D Stack will read into the field on the Synthesis of Solutions card, and use to make the subsystem cards.

- Select "Create Reference Report" from the Dictionary Menu

Click on "OK" in this dialog box, after adding a unique identifier to the default "Reference Report"



You are now ready to Quit the Design/IDEF program.

- Choose "Save" from the File Menu

- Choose "Quit" from the File Menu

You are now back in the D, P & D Stack, viewing the Synthesis of Solutions card.
For information on importing the subsystem list you just created, click on the Help button.

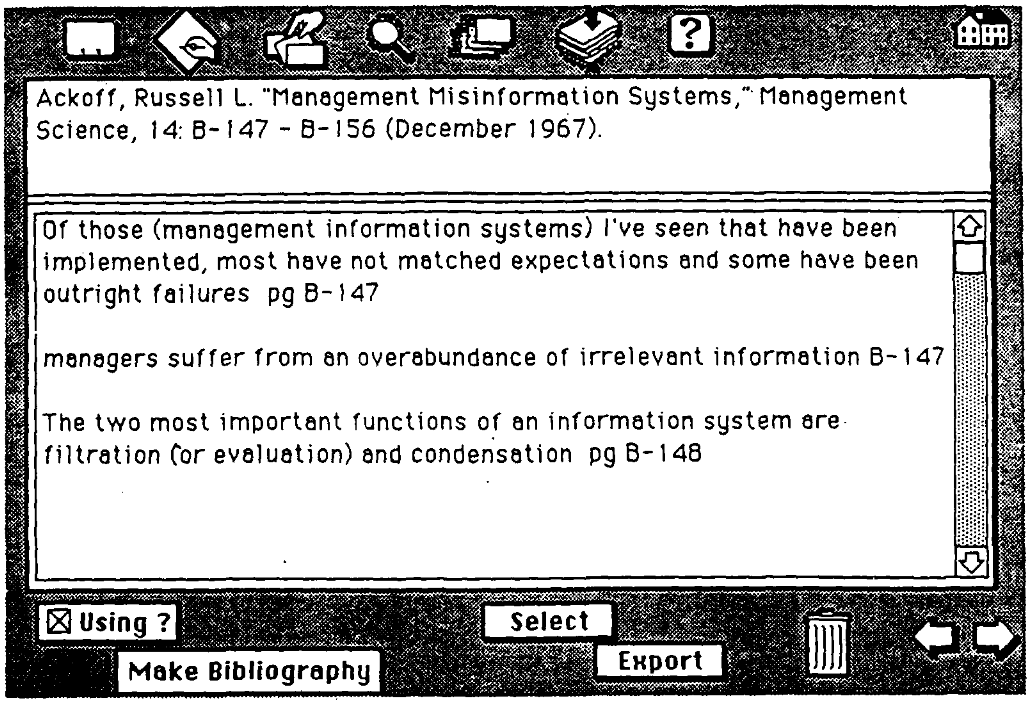
Appendix C : *Bibliography Stack*

Prior to starting work on the HyperCard® stack described in this thesis, some preliminary, smaller projects were completed in order to learn how to work in the HyperCard® environment. The Bibliography Stack was one of those projects.

This stack allows the AFIT student to keep track of references, take notes on these references, and generate bibliographies and notes in text file format. The student may keep all references used throughout an AFIT career in this stack, and generate bibliographies for specific assignments by annotating references with the "Using" button.

The Bibliography stack consists primarily of **Entry Cards,** analogous to the note cards on which many people make notes about sources consulted during the course of a research project.

A sample Entry Card is shown below:



Entry Card Action Button Descriptions :

 — **Card**

This button adds an Entry Card to the stack.

 — **Entry**

This button prompts the user for the data required to fill in a bibliography entry. The user first specifies if the entry is a book or a periodical. If the entry is for another type of sorce, the user will need to fill in the entry manually.

Once the user has specified the source type, a series of dialog boxes query the user for specific information. This information is filled into the entry card, and punctuation is added according to AFIT style guide requriements.

 — **Sort**

This button sorts the entry cards alphabetically by author last name.

 — **Search**

This button locates a specific entry, word, or string of words.

 — **Show**

This button shows each Entry Card in rapid succession.

 — **Compact**

This button eliminates the free space in the stack, thus making operations more efficient.

? — **Help**

This button accesses a set of Help Cards which describe each Entry Card button.

☒ Using ? — **Include in Next Bibliography**

This button allows the user to indicate that an entry should be included in the next bibliography generated.

If an entry is to be included in the next bibliography,the user clicks on this button and an "X" appears in the small box. To remove the entry, the user clicks on the box again, the "X" is removed, and the entry is not included in the next bibliography.

Make Bibliography — **Make a Bibliography**

This button sorts the Entry Cards according to author's last name and writes the entries designated as described above into a text-only file.

Select

Export — **Select** and **Export**

The Select Button allows the user to select text within the scrolling "notes" field and save it for later export to a text file. The author's last name and the date of the reference will be saved along with the selected text.

The Export Button writes the previously selected text to a text-only file.

 — **Delete**

This button deletes the displayed Entry Card.

Stack Information

A copy of the Bibliography Stack can be obtained by contacting:

LtCol Richard Peschke
AFIT/LS
Wright Patterson AFB, Ohio 45433
(513) 255-4437
Autovon 785-4437

Bibliography

Ackoff, Russell L. "Management Misinformation Systems," *Management Science*, *14*: B147-B156 (December 1967).

Adams, Elizabeth Byrne. Professor of Management, George Washington University. Speech. Washington DC, 7 May 1980.

Andrews, William C. "Prototyping Information Systems," *Journal of Systems Management*, *34*: 16-18 (September 1983).

Apple Computer Inc. *HyperCard® User's Guide*. Cupertino CA: Apple Computer Inc., 1987a.

-----. *Apple® HyperCard® Script Language Guide: The HyperTalk™ Language*. Reading MA: Addison-Wesley Publishing Company, Inc., 1987b.

-----. *Technical Introduction to the Macintosh® Family*. Reading MA: Addison-Wesley Publishing Company, Inc., 1987c.

Asimow, Morris. *Introduction to Design*. Englewood Cliffs NJ: Prentice Hall, Inc., 1962.

Booth, Grace M. *The Design of Complex Information Systems: Common Sense Methods for Success*. New York: McGraw-Hill Book Company, 1983.

Bowman, E. H. "Consistency and Optimality in Managerial Decision Making," *Management Science*, *9*: 310-321 (January 1963).

Bryce, Milton. "The IRM Idea," *Datamation*, *33*: 89-92 (April 1987).

-----. "Information Resource Mismanagement," *Infosystems*, *30*: 88-92 (February 1983).

Bush, V. "As We May Think," *Atlantic Monthly*, *176*: 101-108 (July 1945).

Cleveland, Harlan "Information as a Resource," *The Futurist*, *16*: 34-39 (December 1982).

Colter, Mel A. "A Comparative Examination of Systems Analysis Techniques," *MIS Quarterly*, *8*: 51-66 (March 1984).

Conklin, Jeff. "A Survey of Hypertext," MCC Technical Report, Number STP-356-86, Rev.2 (December 1987).

Davis, Gordon B. "Strategies for Information Requirements Determination," *IBM Systems Journal*, *21*: 4-30 (1982).

Davis, Gordon B. and Margarethe H. Olson. *Management Information Systems, Conceptual Foundations, Structure, and Development (Second Edition)*. New York: McGraw Hill Book Co., 1985.

Davis, Michael W. *Applied Decision Support*. Englewood Cliffs NJ: Prentice Hall, 1988.

Diebold, John. *Managing Information: The Challenge and the Opportunity*. New York: AMACOM, 1985.

Eden, C. et al. *Thinking in Organizations*. London: Macmillan Press Ltd., 1979.

Goodman, Danny. *HyperCard™ Developer's Guide*. New York: Bantam Books, 1988.

-----. *The Complete HyperCard™ Handbook*. New York: Bantam Books, 1987.

Gowin, D. Bob and Novak, Joseph D. Lecture materials - Improving College Teaching. Cornell University, Ithaca NY, June 1982.

Hall, Arthur D. *A Methodology for Systems Engineering*. Princeton NJ: D. VanNostrand Company Inc., 1962.

Hoffman, Eliahu. "Defining Information: An Analysis of the Information Content of Documents," *Information Processing and Management*, *16*: 291-304 (1980).

Kaehler, Carol. *HyperCard™ Power: Techniques and Scripts*. Reading MA: Addison-Wesley Publishing Company, Inc., 1988.

-----. *Macintosh® Plus User's Manual*. Cupertino CA: Apple Computer, Inc., 1986.

Kaufmann, R. A. and F. W. English. *Needs Assessment: Concept and Application*. Englewood Cliffs NJ: Educational Technology Publications, 1979.

Keen, P. G. and G. R. Wagner. "DSS: An Executive Mind-Support System," *Datamation*, *25*: 117-122 (November 1979).

Land, F. F. "Adapting to Changing User Requirements.," *Information and Management*, *5*: 59-75 (1982).

Lederer, Albert L. "Information Requirements Analysis," *Journal of Systems Management*, *32*: 15-19 (December 1981).

Martin, Merle P. "The Human Connection in System Design Part VI - Designing Systems for Change," *Journal of Systems Management*, *38*:14-18 (July 1987).

McFarren, Michael R. *Using Concept Mapping to Define Problems and Identify Key Kernels During the Development of a Decision Support System*. MS thesis, AFIT/GST/EN/87J-12. School of Engineering, Air Force Institute of Technology (AU), Wright Patterson AFB OH, June 1987.

Montazemi, A. R. and D. W. Conrath. "The Use of Cognitive Mapping for Information Requirements Analysis," *MIS Quarterly*, *10*: 45-55 (March 1986).

Naisbitt, John. *Megatrends: Ten New Directions Transforming Our Lives*. New York: Warner Books, Inc., 1982.

Novak, Joseph D. and D. Bob Gowin. *Learning How to Learn*. New York: Cambridge University Press, 1984.

Ostrofsky, Benjamin, Professor, University of Houston. Personal interview. 23d Annual International Logistics Symposium, Orlando FL, 23 August 1988.

-----. *Design, Planning and Development Methodology*. Englewood Cliffs NJ: Prentice Hall Inc., 1977.

Ostrofsky, Benjamin and E. A. Kiessling. "A Review and Analysis of IBM Business System Planing (BSP) and an Information Resource Management Application," Working Paper Series, University of Houston, University Park, Houston, Texas. (August 1984).

Peschke, Richard E. "A Structured Optimization Method for Information Resources Management," PhD Dissertation, College of Business Administration, University of Houston (December 1985).

Peschke, Richard E. and Kathleen M. Austin. ""Hypertext: The Logistician's Answer to Information Overload?," *Proceedings of the SOLE 23d Annual International Logistics Symposium.* 67-70. Huntsville AL: Society of Logistics Engineers, 1988.

Robey, Daniel and William Taggart. "Human Information Processing in Information and Decision Support Systems," *DSS: A Data Based, Model Oriented, User Developed Discipline*, edited by William C. House. New York: PBI, 1983.

Seagle, John P. and Salvatore Belardo. "The Feature Chart: A Tool for Communicating the Analysis for a Decision Support System," *Information and Management*, *10*:11-19 (1986).

Shafer, Dan. *HyperTalk™ Programming*. Indianapolis: Hayden Books, 1988.

Siegel, Paul. *Strategic Planning of Management Information Systems*. New York: Petrocelli Books, 1975.

Simon, Herbert A. et al. "Decision Making and Problem Solving," *Interfaces*, *17*: 11-31 (September-October 1987).

Simons, Jacob V. Jr. and Benjamin Ostrofsky. "Research in Multiple Criteria Design Optimization," *Proceedings of the SOLE 23d Annual International Logistics Symposium* 49-58. Huntsville AL: Society of Logistics Engineers, 1988.

Smith, John B. and Stephen F. Weiss. "An Overview of Hypertext," *Communications of the ACM, 31*: 816-819 (July 1988).

Snow, Janice and Randall Albright. *Design/IDEF™ User's Manual*. Cambridge MA: Meta Software Corporation, 1987.

Synott, William R., and William H. Gruber. *Information Resource Management: Opportunities and Strategies for the 1980s*. New York: John Wiley and Sons, Inc., 1981.

Tom, Paul L. *Managing Information as a Corporate Resource*. Glenview IL: Scott, Foresman and Company, 1987.

U. S. General Services Administration, Office of Information Resources Management. *FIRMR: Federal Information Resources Management Regulation.* GPO Item No. 580-C-6. Washington: Government Printing Office, 1985.

Valusek, John R. Class lectures in OPER 652, Decision Support Systems. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, October 1988a.

-----. "Adaptive Design of DSSs: A User's Perspective," *Transactions of the Eighth International Conference on DSS.* 105-112. TIMS, 1988b.

Valusek, John R. and Dennis G. Fryback. "Information Requirements Determination: Obstacles Within, Among and Between Participants," *Proceedings of the End-User Computing Conference, Association for Computing Machinery,* 1985.

Vaughn, Tay. *Using HyperCard®: From Home to HyperTalk™*. Carmel IN: Que Corporation, 1988.

Yadav, Surya Bhan. "Determining an Organization's Information Requirements: A State of the Art Survey," *Database, 14*: 3-20 (Spring 1983).

VITA

Captain Tamara C. Mackenthun [redacted] She graduated from high school in San José in 1577, and attended Arizona State University, from which she received a Bachelor of Science Degree in Psychology in 1981. Upon graduation, she was granted a commission in the USAF through the ROTC program, and was called to active duty in October 1981. Her initial duty assignment was to Williams AFB, Arizona, where she attended Undergraduate Pilot Training (UPT). After being medically eliminated from UPT, she was assigned to the AF Office of Special Investigations (OSI) and served as a Special Agent of the OSI at Williams AFB and Seymour Johnson AFB, North Carolina. She cross-trained into the Administration career field in 1983, and was assigned to the 4th Supply Squadron, Seymour Johnson AFB, as the Squadron Section Commander. She was assigned to George AFB, California, in March, 1985, where she served as Adjutant, 562d Tactical Fighter Training Squadron; Commander, 37th Headquarters Squadron; and Squadron Section Commander, 37th Aircraft Generation Squadron. She entered the School of Systems and Logistics at the Air Force Institute of Technology in May, 1987.

Captain Mackenthun is married to Captain Michael L. Mackenthun, USAF, and they have one child, Christopher.

[redacted]

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GIR/LSQ/88D-8	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION School of Systems and Logistics 6b. OFFICE SYMBOL (If applicable) AFIT/LSY	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology (AU) Wright Patterson AFB OH 45433-6583	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

STRUCTURED REQUIREMENTS DETERMINATION FOR INFORMATION RESOURCES MANAGEMENT

12. PERSONAL AUTHOR(S)

Tamara C. Mackenthun, B.S., Captain, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
M.S. Thesis	FROM ____ TO ____	1988 December	101

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
05	01	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

Information, Information Resources Management, Structured Optimization, Requirements Determination, Information Systems

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Thesis Chairman: Richard E. Peschke, PhD, Lt Colonel, USAF
Assistant Professor
Department of Quantitative Management

Approved for public release IAW AFR 190-1.

WMaue

WILLIAM A. MAUER
Associate Dean 27 JAN 1989
School of Systems and Logistics
Air Force Institute of Technology (AU)
Wright-Patterson AFB OH 45433-6583

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Richard E. Peschke, Lt Colonel, USAF	22b. TELEPHONE (Include Area Code) (513) 255-6280 22c. OFFICE SYMBOL AFIT/LSQ

DD Form 1473, JUN 86 *Previous editions are obsolete.* SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

The purpose of this research was to provide the Information Resources Management System designer with a framework on which to structure the decisions which must be made in order to translate rapidly changing information needs into plans for Information Resources Management Systems which implement rapidly changing technology.

The HyperCard programming environment and the Design/IDEF diagramming tool were used to develop a design support system which guides the Information Resources Management (IRM) system designer through the requirements determination stage of Dr. Benjamin Ostrofsky's Design, Planning and Development Methodology. This system consists of the Design, Planning and Development (D, P & D) Stack, a Help stack, and a User's Manual. The system guides the IRM system designer through the requirements determination process, assists in the collection of data and organizes that data into a form which can be subjected to objective analysis and optimization.

The system currently supports only the requirements determination phase of a complete Information Resources Management System design methodology. It is intended to serve as input to future development of a complete system to assist the Information Resources Management System designer with all phases of the design process.